National Research Council Canada
Conseil national de recherches Canada

Division of Electrical Engineering
Division de génie électrique

ERB-995

A Computer Vision System for Matching 3-D Range Data Objects

N.N. Abdelmalek

September 1987

Canada

NRCC No. 28053

88-04440/

Copyright 1987 by
National Research Council of Canada

Permission is granted to quote short excerpts and to reproduce figures and tables from this report, provided that the source of such material is fully acknowledged.

Copyright 1987 par
Conseil national de recherches du Canada

Il est permis de citer de courts extraits et de reproduire des figures ou tableaux du présent rapport, à condition d'en identifier clairement la source.

Additional copies are available free of charge from:

Editorial Office, Room 301
Division of Electrical Engineering
National Research Council of Canada
Ottawa, Ontario, Canada
K1A 0R8

Des exemplaires supplémentaires gratuits peuvent être obtenus à l'adresse suivante:

Bureau des publications, Pièce 301
Division de génie électrique
Conseil national de recherches du Canada
Ottawa (Ontario) Canada
K1A 0R8

A COMPUTER VISION SYSTEM FOR MATCHING 3-D RANGE DATA OBJECTS

N.N. Abdelmalek

ERB-995
NRCC No. 28053

SEPTEMBER 1987

TABLE OF CONTENTS

	Page
ABSTRACT/RÉSUMÉ	v
1. INTRODUCTION	1
2. DESCRIPTION OF THE SYSTEM	3
2.1 Data Forms	3
2.2 Reducing the Processing Time	4
2.3 Noise, Inherited and Added	4
2.4 Data Smoothing	4
2.5 Further Reducing the Processing Time	4
2.6 The Contour of the Object Base Silhouette	5
3. CALCULATING THE INVARIANT PARAMETERS	5
4. EXPERIMENTAL RESULTS AND COMMENTS	9
ACKNOWLEDGEMENTS	14
REFERENCES	14
APPENDIX	17
Driver DRSHOW	17
Driver DRMATCH	25
Subroutine DR3DRANGE	29
Subroutine READIMAGE	44
Subroutine READIMAG2	46
Subroutine BOUNDS	49
Subroutine READSTRNG	52
Subroutine OBJECT	54
Subroutine OBJECTROT	62
Subroutine SMOOTHIMG	66
Subroutine DISPLAY	68
Subroutine MOMENTAXS	70
Subroutine EQUATION	72
Subroutine NORMEQLS	74
Subroutine ELLIPSE	77
Subroutine KONTOURS	80
Subroutine MMTAXSKNT	83
Subroutine CONVOL5	85
Subroutine SGNMAP	95
Subroutine MOMENTS	97

FIGURES

1. The seven objects scanned by the laser range camera 9

2. Object 1, scanned in four differential oriented positions 12

ABSTRACT

An efficient global system for matching 3-D range data objects is here presented. The obtained invariant parameters used for matching purposes are functions of the object centroid coordinates, its volume, its base area, second-order central moments and standard deviations for the object surface and for the contour of the object base silhouette. We use a simple method to restrict our matching process to the model objects of the same surface type (convex or concave). A separate set of invariant parameters are calculated and stored into the matching library for each stable position for each used object. The matching system is tested on a number of objects including simulated ones, with and without added noise and was found to be fast and robust.

RÉSUMÉ

On présente un système global efficace de couplage d'objets pour lesquels on possède des données tridimensionnelles de profondeur. Les paramètres invariants utilisés pour le couplage sont fonction des coordonnées du centre de gravité de l'objet, de son volume, de l'aire de sa base, des moments centraux de deuxième order et des écarts-types pour la surface de l'objet et pour le contour de la silhouette de la base de l'objet. Nous utilisons une méthode simple pour limiter notre processus de couplage aux objets modèles ayant le même type de surface (convexe ou concave). Pour chaque position stable de chaque objet utilisé, on détermine par calcul un ensemble distinct de paramètres invariants et on l'emmagasine dans la bibliothèque de couplage. Le système de couplage est mis à l'essai sur un certain nombre d'objets, y compris des objets simulés, avec et sans bruit ajouté, et on a trouvé qu'il est rapide et puissant.

A COMPUTER VISION SYSTEM FOR MATCHING THREE-DIMENSIONAL RANGE DATA OBJECTS

N.N. Abdelmalek

1. INTRODUCTION

Recognition and matching of three-dimensional (3-D) objects from camera views is an important problem in image recognition. It has wide application, particularly for robotic vision systems.

Many of the 2-D object recognition methods are extended to the recognition of 3-D objects, such as the Fourier descriptors method (1, 2) and the method of invariant moments (3-7). Maitra (7) pointed out that there are typographical errors in some moment parameter expressions in references 3-5 and other references.

We have experimented with the method of invariant moments in which we calculate seven normalized central invariant moment parameters (7) for the object surface. These parameters are invariant under object translation, plane rotation, and a change in scale. We discovered that for the objects we used, the five third-order moment parameters are very small. They are of the order of, or less than, the round-off error level of the computer and could not be considered invariant. We then calculated the seven (unnormalized) central invariant moment parameters (7). They are invariant under translation and plane rotation but not under a scale change. We again found out that for the objects we used, the five third-order moment parameters are very large and of very wide range. Again they could not be used as matching parameters.

We also experimented with techniques using the object surface Gaussian and mean curvatures (8, 9). We again found that, for the objects we used, the Gaussian and mean curvatures are very small and sensitive to noise. There are also difficulties concerning the object sudden edges. We abandoned such techniques.

Our approach here is based on making use of the second-order central moments and the standard deviations of the object surface and other invariant parameters of the object. We also use the second-order central moments and standard deviation of the contour of the object base silhouette.

Moreover, we fit the object surface in a least-squares sense with a paraboloid (10), elliptic or hyperbolic. The intersection of this

paraboloid with the plane z = 0 is either an ellipse or a hyperbola, denoted by the primary ellipse or hyperbola, or simply the primary conic. The tested objects are thus divided into two groups. The objects with convex surfaces are fitted with elliptic paraboloids and those with concave surfaces are fitted with hyperbolic paraboloids. We restrict our matching process to the object models of the same surface type: convex or concave.

In this respect, Wang, Magee, and Aggarwal (11) presented a method for matching 3-D objects using silhouettes. The volume and the three normalized principal second-order moments defined with respect to three principal axes are computed for the different principal silhouettes for the 3-D objects. These parameters are stored and used for matching purposes.

Rather than calculating the principal moments using different silhouettes for the 3-D object (11), we calculate a variety of second-degree central moments. Also, rather than using a binary representation of the image (6, 11), we treat the z-coordinates of the object as the pixel values in a grey scale image.

For matching purposes we calculate the following nine invariant parameters: (1) the volume of the object, (2) the object base area, (3) the standard deviation of the object surface heights, (4) the maximum or the minimum moment of inertia of the object surface about axes through the point $(x_{mean}, y_{mean}, 0)$, where (x_{mean}, y_{mean}) are the (x, y)-coordinates of the object surface centroid, (5) the standard deviation of the (x, y)-coordinates (12) of the object surface, (6) the maximum or the minimum moment of inertia of the contour of the object base silhouette, about axes through the contour centroid, (7) the standard deviation of the (x, y)-coordinates of the contour of the base silhouette, (8) the semi-major axis or the semi-minor axis of the primary conic mentioned above, and (9) the eccentricity of the primary conic. The eccentricity is <1 if the primary conic is an ellipse and is >1, if it is a hyperbola. The value of the conic eccentricity is used as the ninth matching parameter.

The matching parameters (1) to (8) are invariant under translation and plane rotation but not under a change of scale. The conic eccentricity is invariant under a change in scale.

Nevertheless, we here mention an important point. That is that the range data we get from the range camera are already calibrated as we get the (x, y, z)-coordinates of the object surface. Therefore we need no longer worry about the position of the camera with relation to the object the camera scannned.

We point out that the contour of the object base is obtained if the

object is given in a matrix form only. As a result, the parameters (6) and (7) could not be calculated if the object is originally given in a string form. See Section 2 below.

The angle that the axis of maximum (minimum) moment of inertia of the object surface makes with the I-axis is easily calculated and may be used for picking purposes in a robot vision system. Also, the angle that the axis of maximum (minimum) moment of inertia of the base silhouette contour makes with the I-axis may be used as an alternative for picking purposes.

A separate set of invariant parameters are calculated and stored for the matching library for each stable position for each used object.

This matching system has been tested on seven given objects and two simulated ones, with and without added Gaussian noise, and is found to be fast and robust.

In Section 2 the matching system is described. In Section 3, the outline of the calculation of the invariant parameters is presented. In Section 4, experimental results are displayed and comments and conclusion are given.

2. DESCRIPTION OF THE SYSTEM

2.1 Data Forms

Recently a new design for a laser range image camera was proposed in the Division of Electrical Engineering of the National Research Council of Canada (13-15). It provides a means to obtain a very large field of view without compromising on resolution. Besides, it gives a large increase in speed of measurement and the shadow effects are reduced considerably.

The range data from the camera are given in one of the two forms: (a) A Real*4 (R*4) matrix of size 256 × 256 whose (i, j) value is the z-coordinate at point (i, j) of the range object. We denote this by the matrix representation of the object. (b) An R*4 string of coordinates in the form $\{x(1), y(1), z(1), x(2), y(2), z(2), \ldots, x(k), y(k), z(k)\}$, which are successively the x-, y-, and z-coordinates of the successive points of the object. For this data, the x- and y-coordinates in general are not at equal intervals. We denote this by the x-y-z-string vector presentation of the object surface.

In the present work, a computer program is designed to read the above two forms of files as well as two other forms for simulated objects. These

are Integer*4 (I*4) and R*4 128 × 128 matrices. However, here we mainly describe the matching system assuming the input data are given in a 256 × 256 matrix form.

2.2 Reducing the Processing Time

To reduce the processing time of the matching system to about one quarter of its amount, when a 256 × 256 range object matrix is read, the image matrix is immediately reduced to a 128 × 128 one. This is done by taking every second matrix element, every second row. The invariant parameters mentioned above are calculated for the 128 × 128 object matrix obtained.

Surfaces for simulated objects are created by one of the subroutines in the form of R*4 128 × 128 matrices. These are for simple symmetric geometrical objects of different sizes such as spheres, ellipsoids, and paraboloids of revolution and cylinders.

2.3 Noise, Inherited and Added

When the object matrix is reduced from a 256 × 256 to a 128 × 128 matrix, the matrix obtained is noisier than the original 256 × 256 one. Yet, to test our system, Gaussian noise of zero mean and standard deviation σ_n may still be added to the object matrix, for the given or the simulated objects.

2.4 Data Smoothing

The range data contaminated with noise, either inherited or added, may be smoothed first before processing. Smoothing is done by taking the average for each pixel element in the 128 × 128 matrix over a 3 × 3, 5 × 5, or 7 × 7 window. The computation is then done on the smoothed matrix. The smoothing operation can only be done if the object data are given in a matrix form.

We here mention that we should have smoothed the given 256 × 256 data object matrix before reducing it to a 128 × 128 matrix. Howerver, we chose to reduce the given data first and then smooth it. The reason is that the smoothing operation is very time-consuming, and it is faster to smooth the 128 × 128 matrix rather than the 256 × 256 one.

2.5 Further Reducing the Processing Time

Again, if the image is given in a matrix form, not in the x-y-z-string form, to further reduce the processing time the following parameters and

files are obtained and used. (a) The four (i, j)-coordinates of the corners of the rectangle inside which the nonzero (i, j) pixel values are enclosed. We denote this by the enclosing rectangle of the object. (b) R*4 x-, y-, and z-vectors, respectively, for the i-, j-, and z- coordinates of the surface at hand are obtained. That is, given an object surface in a matrix form, we get the object surface in a set of three string forms, which we denote by the x-, y-, and z-string vectors of the object. Each string is of length k, which is the number of the surface points of the object or the number of nonzero elements of the object matrix.

Due to the presence of noise in the object data, by a zero matrix element we mean an element whose value is less that $z_{max}/100$, where z_{max} is the largest element in the matrix. The parameter 100 is taken arbitrarily. If the data are too noisy, the parameter 100 may be replaced by 50.

A 128 × 128 object matrix contains 16 384 elements. Yet the number of nonzero matrix elements k may range from about 2000 to 6000 elements. We do our computation inside the enclosing rectangle of the object and/or using the three k-string vectors of the object surface, with a substantial reduction of the processing time. The processing time is reduced by the ratio k, the number of nonzero matrix elements to the total matrix elements.

2.6 The Contour of the Object Base Silhouette

By the contour of the object base silhouette, we mean the outside and also the inside contours of the two-dimensional object base. If the 3-D object is hollow, the base will show holes inside its boundaries. The inside contours are the contours of these holes. Such contours can only be obtained if the object is given in a matrix form.

These contours are obtained by a simple technique. The object matrix is scanned column-wise (row-wise) inside the enclosing rectangle. Every non-zero element of this matrix (non-zero object surface z-coordinate) is considered. Each of the eight neighbouring elements to this non-zero element as a center is examined. If any of the eight elements is a zero element, the center element is considered a boundary or a contour point. Each of the elements of the base contours is given a character 1 and each of the rest of the base elements is given the blank character ' '.

3. CALCULATING THE INVARIANT PARAMETERS

The significance of the invariant parameters is obvious from their values as calculated in this section.

1. The object volume is none other than the sum of the k elements in the z-string vector of the object surface.

2. The object base area is clearly the length k of any of the three string vectors of the object surface. The base area thus has an integer value.

The coordinates of the surface centroid (x_{mean}, y_{mean}, z_{mean}) are given respectively by

$$x_{mean} = \sum_{i=1}^{k} x_i/k$$

[1] $$y_{mean} = \sum_{i=1}^{k} y_i/k$$

$$z_{mean} = \sum_{i=1}^{k} z_i/k$$

3. The standard deviation σ_z of the z-coordinates of the surface of the object is given by

[2] $$\sigma_z^2 = \sum_{i=1}^{k} (z_i - z_{mean})^2/k$$

As the object is rotated about its z-axis, the shadow effect results in some variations in the values of the calculated volumes, base areas, and the standard deviations of the z-coordinates.

4. The maximum (minimum) central moments of inertia of the object surface about axes through the point (x_{mean}, y_{mean}, 0), are a measure of the elongation of the object along the proper axis. These are given by the following formulas (16).

[3a] $$2I_{max} = (I_{xx} + I_{yy}) + [(I_{xx} - I_{yy})^2 + 4I_{xy}^2]^{1/2}$$

$$2I_{min} = (I_{xx} + I_{yy}) - [(I_{xx} - I_{yy})^2 + 4I_{xy}^2]^{1/2}$$

and the direction of the principal axes is given by

[3b] $$\tan 2\theta = 2I_{xy}/(I_{yy} - I_{xx})$$

If $I_{xy} = 0$, $2\theta = 0°$ or $180°$ and if $I_{xx} = I_{yy}$, $\theta = \pi/4$.

In [3a,b],

$$I_{xx} = \sum_{i=1}^{k} z_i (y_i - y_{mean})^2$$

[3c] $$I_{yy} = \sum_{i=1}^{k} z_i (x_i - x_{mean})^2$$

$$I_{xy} = \sum_{i=1}^{k} z_i (x_i - x_{mean})(y_i - y_{mean})$$

5. The standard deviation of (x, y)-coordinates of the object surface is a parameter used by Alt (12) and is given by

$$\sigma_{xy} = (\sigma_x^2 + \sigma_y^2)^{1/2}$$

where

[4] $$\sigma_x^2 = I_{yy}/k$$
$$\sigma_y^2 = I_{xx}/k$$

This parameter σ_{xy} is proportional to the square root of the first second-order invariant moment parameter $\phi(1)$, (reference 7, p. 697), which is obviously an invariant parameter.

6. The maximum and the minimum central moments of inertia of the contour of the base silhouette are calculated from the same formulas [3], with the z_i replaced by 1. The summation in [3] would be taken over the number of points of the base contour silhouette.

7. The formula for the standard deviation of the (x, y)-coordinates of the contour of the base silhouette is similarly obtained from formula similar to eq. [4].

8. The semi-major, semi-minor axes of the primary conic and its eccentricity are described in the next sub-section.

9. Fitting elliptic (hyperbolic) paraboloid: Fitting the object surface in the least-squares sense by a paraboloid was suggested recently by Grosky and Jain (10). They considered only the elliptic case and their method of calculating the conic parameters is different from ours. The values of the conic parameters are sensitive to the shadow effects and the calculated values for the same object with different translation

and/or orientations vary substantially. See the experimental results in Section 4. An advantage in calculating the eccentricity is in finding out whether the object surface is convex or concave.

The equation of the fitting paraboloid may be given by

$$[5a] \qquad z = a_1 + a_2X^2 + a_3XY + a_4Y^2$$

where

$$[5b] \qquad \begin{aligned} X_i &= x_i - x_{mean} \\ Y_i &= y_i - y_{mean} \end{aligned}$$

and again (x_{mean}, y_{mean}) are the (x, y)-coordinates of the surface centroid.

Substituting the k coordinates x, y, and z of the three k-string vectors into eqs. [5a,b] we get the system of linear equations

$$[6a] \qquad \begin{bmatrix} 1 & x_1^2 & x_1y_1 & y_k^2 \\ 1 & x_2^2 & x_2y_2 & y_k^2 \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ 1 & x_k^2 & x_ky_k & y_k^2 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ \\ \\ \\ z_k \end{bmatrix}$$

In matrix-vector form, eq. [6a] may be written as

$$[6b] \qquad Ca = f$$

where C is the coefficient matrix in [6a], f is the right-hand side vector, and $a = (a_1, a_2, a_3, a_4)$ is the unknown vector.

The least-squares solution vector $\{a\}$ is the solution of the normal equation

$$[7] \qquad C^TCa = C^Tf$$

where C^T is the transpose of matrix C.

In our experimental results, matrix C^TC in [7] is a well-conditioned matrix. Vector $\{a\}$ is obtained by first factorizing C^TC by Cholesky's decomposing method.

By rotating the I-J axes an angle θ,

$$\tan 2\theta = \frac{a_3}{(a_2 - a_4)}$$

the equation of the paraboloid reduces to

$$z = a_1' + a_2'x^2 + a_4'y^2$$

where the $\{a_i'\}$ parameters are easily calculated from the $\{a_i\}$ and θ. The equation of the primary conic is thus

$$0 = a_1' + a_2'x^2 + a_4'y^2$$

from which the semi-major, semi-minor axes and the eccentricity of the basic ellipse (hyperbola) are obtained.

4. EXPERIMENTAL RESULTS

We experimented with seven given objects and two simulated ones. Figure 1 shows one view of each of the seven given objects. One of the simulated objects is an ellipsoid of revolution and the other one is a paraboloid of revolution. Figure 2 shows object 1 in four different orientations.

Each of objects 1, 2, and 8 is translated and rotated an arbitrary angle about the object z-axis four times. Each of the other objects is translated and rotated five times. As soon as the matrix for the given object is read it is reduced to an 128 × 128 matrix. Gaussian noise of zero mean and standard deviation 2.0 is added to the matrix elements of all objects. Adding the noise may result in some negative matrix elements. These elements are given zero values. In our experiments, the object matrix has not been smoothed.

Table 1 shows the calculated invariant parameters (1) - (9) for the nine objects. The left-hand numbers (1), (2), ..., refer to the object number. The first column to the left gives the surface type; E stands for elliptic or convex and H for hyperbolic or concave surface. Column 2 gives the eccentricity of the primary conic. Columns 3 and 4 give the object

Figure 1. The seven objects scanned by the laser range camera

volume and base area, respectively. Column 5 shows the semi-major axis of the primary conic. Column 6 gives the standard deviation of the surface z-coordinates. Column 7 shows the surface maximum moment of inertia about an axis through the point (x_{mean}, y_{mean}, 0) and column 8 shows the standard deviation of surface (x, y)-coordinates. Columns 9 and 10 are the corresponding columns to columns 7 and 8 for the contour of the base silhouette.

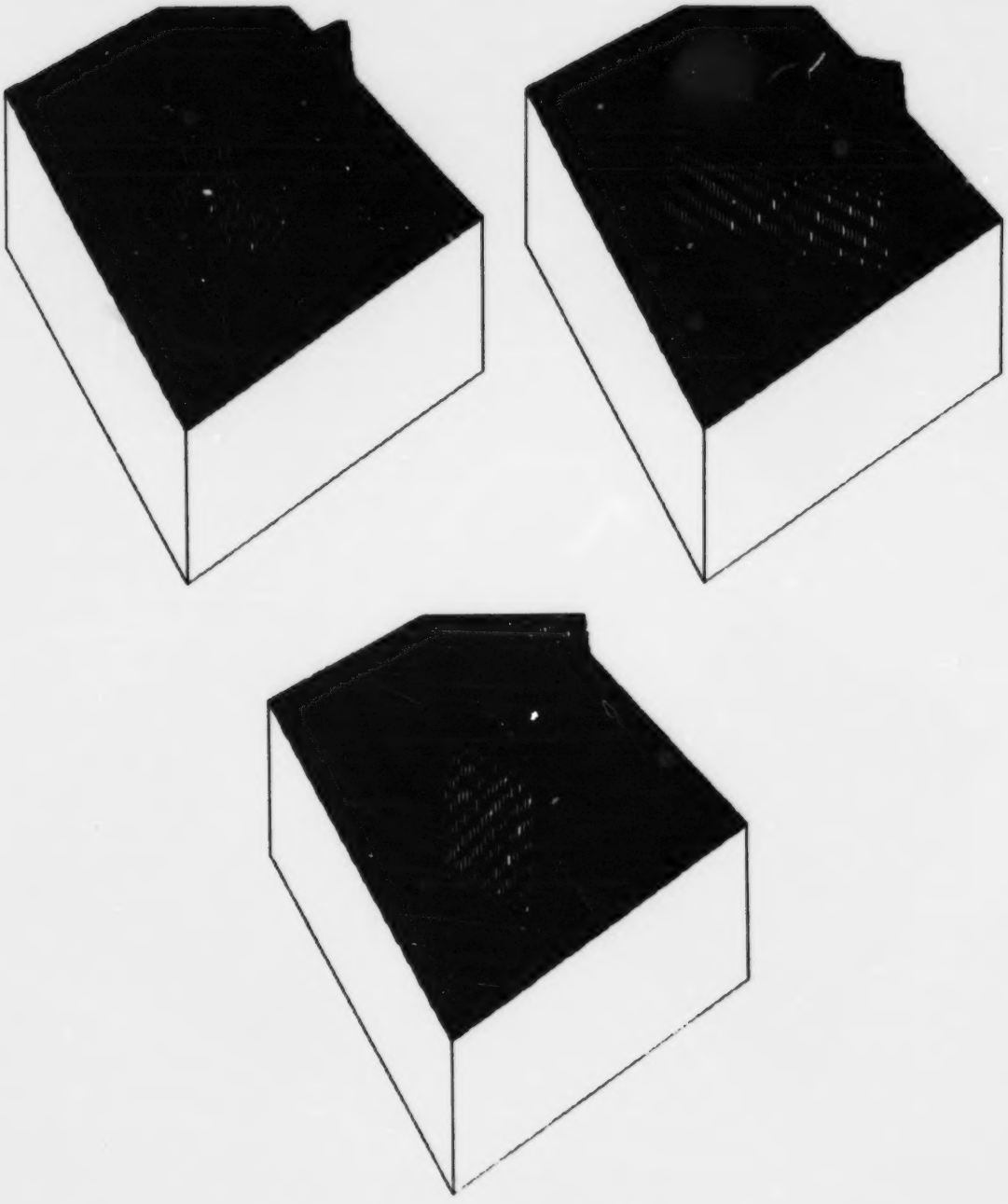

Figure 1. Concluded

Table 1 shows that the results of the simulated objects, objects 8 and 9, are very consistent and do not reflect the variations in the results due to the shadow effects and the inherited noise of the real objects. We shall thus concentrate on discussing the results for the real objects.

Figure 2. Object 1, scanned in four differential oriented positions

Table 1 shows the clustering of the parameter values for the same object with different orientations. The results indicate that the smallest percentage variations in the parameters exist for the five parameters: the object volume, its base area, the standard deviation for the object surface z-coordinates, the standard deviation for the object

Table 1. Calculated invariant parameters

Object	Ind	Eccentricity	Volume	Base area	Semi-major	S.D. S. z	S. Max. M.O.I.	S.D. S. x-y	C. Max. M.O.I.	S.D. C. x-y
(1)	E	0.57	100398	5990	52.98	11.95	38013956	26.15	508255	40.19
	E	0.74	105142	6176	59.40	11.78	46743272	26.63	423596	44.94
	E	0.48	100074	5973	51.34	11.98	34360592	26.04	469034	40.25
	E	0.66	100526	6064	55.61	11.93	40805140	26.27	544341	40.93
(2)	H	1.13	38487	3266	60.91	5.01	11177562	24.06	165993	32.23
	H	1.14	39212	3306	66.21	5.08	12272569	24.10	173829	32.30
	H	1.13	38897	3230	74.60	5.19	13954977	24.13	190729	32.12
	H	1.31	40943	3304	47.37	5.33	14887268	24.33	153724	32.05
(3)	H	1.38	32300	2859	32.68	5.54	15309412	25.14	198692	27.46
	H	1.37	32158	2827	31.79	5.52	15196126	25.11	167457	27.18
	H	1.42	32200	2868	33.59	5.53	15713639	25.28	182175	27.15
	H	1.39	32366	2866	32.70	5.51	15376374	25.16	202442	27.82
	H	1.42	32434	2887	31.48	5.59	15710798	25.30	205937	27.39
(4)	E	0.90	38872	2639	84.21	7.58	7846435	18.88	194034	29.82
	E	0.98	38328	2594	174.47	7.58	7845173	18.82	223812	29.24
	E	0.97	38234	2598	146.27	7.61	7640668	18.78	217750	29.29
	E	1.00	38155	2580	378.75	7.59	7748164	18.76	215918	28.88
	E	0.98	38264	2598	195.82	7.64	7597389	18.74	221919	29.27
(5)	E	0.84	39564	2769	57.49	6.53	13319996	21.26	238066	31.83
	E	0.84	39345	2749	57.44	6.48	12948094	21.11	217935	31.19
	E	0.84	38627	2734	56.70	6.59	12727911	21.12	217896	31.22
	E	0.84	38761	2685	57.68	6.43	12655618	21.01	205801	31.21
	E	0.84	39356	2754	56.60	6.53	12883035	21.10	232355	31.37
(6)	E	0.66	63962	3708	74.55	6.06	32173478	25.87	303936	35.78
	E	0.68	63915	3727	73.31	6.18	31496898	25.74	307227	35.67
	E	0.76	63301	3717	76.78	6.12	31402314	25.76	293759	35.59
	E	0.62	63407	3699	75.00	5.92	32299624	25.95	273538	36.25
	E	0.77	63091	3707	77.46	6.08	30632904	25.62	267421	35.50
(7)	H	2.25	63576	2455	31.47	8.44	28406098	23.24	183349	29.48
	H	2.22	63412	2451	33.00	8.37	27906054	23.12	179881	29.41
	H	2.19	63754	2479	28.45	8.54	27917906	23.13	174070	29.44
	H	2.14	62563	2404	36.94	8.19	28152192	23.24	164927	29.65
	H	2.12	63656	2477	27.78	8.59	27110728	22.93	152931	29.36

Table 1. Calculated invariant parameters (Concluded)

Object	Ind	Eccentricity	Volume	Base area	Semi-major	S.D. S. z	S. Max. M.O.I.	S.D. S. x-y	C. Max. M.O.I.	S.D. C. x-y
(8)	E	0.65	149504	4222	66.80	5.55	60419772	25.28	238124	36.47
	E	0.65	149504	4222	66.80	5.55	60419712	25.28	241943	36.22
	E	0.65	149504	4222	66.80	5.55	60419788	25.28	238056	36.47
	E	0.65	149504	4222	66.80	5.55	60419692	25.28	244998	36.28
(9)	E	0.83	37426	2383	51.86	6.15	11758341	20.23	171995	29.03
	E	0.83	37426	2383	51.86	6.15	11758336	20.23	172334	28.95
	E	0.83	37426	2383	51.86	6.15	11758348	20.23	147124	29.27
	E	0.83	37426	2383	51.86	6.15	11758341	20.23	171995	29.03
	E	0.83	37426	2383	51.86	6.15	11758347	20.23	147196	29.15

(x, y)-coordinates, and that of the base silhouette contour. We denote these parameters by set 1. This makes set 1 the most reliable parameters for the matching purposes. More percentage variations occur in the maximum moments of inertia of the object surface and the maximum moments of inertia of the contour of the base silhouette, which we denote by set 2. The least reliable parameters are the primary conic eccentricity and semi-major axis.

Nonetheless, we have given each parameter the same weight. Matching is done by counting the largest number of times the object parameters are closest, percentagewise, to those of the model.

The experimental results show that this matching system is fast and robust.

ACKNOWLEDGEMENTS

The author had stimulating discussions with Nestor Burtnyk, Shadia Elgazzar, and Mark Rioux. Robert Kadamani made this system an interactive one on the IIS display. Kenneth Evans, Arthur Binch, and Michael Duggan helped with the IIS display, the VAX system, and the Camera. I am deeply grateful to all of them. A special thanks is extended to Luc Cournoyer for providing the Range Data.

I am also grateful to Evelyn Kidd and Theresa Tierney, staff of the Editorial Office of the Division of Electrical Engineering, for their care in producing this and my previous reports.

REFERENCES

1. C.T. Zahn and R.Z. Roskies. Fourier descriptors for plane closed curves. IEEE Trans. Comput. C-21: 269-281; 1983.

2. C.W. Richard and H. Hemami. Identification of three-dimensional objects using Fourier descriptors of the boundary curve. IEEE Trans. Pattern Anal. Mach. Intell. PAMI-2: 127-136; 1980.

3. M.-K. Hu. Visual pattern recognition by moment invariants. IRE Trans. Inf. Theory, IT-8: 179-187; 1962.

4. R.Y. Wong and E.L. Hall. Scene matching with invariant moments. Comput. Vision, Graphics, Image Process. 8: 16-24; 1978.

5. E.L. Hall. Computer Image Processing and Recognition. Academic Press. 1979.

6. S.A. Dudani, K.F. Breeding, and R.B. McGee. Aircraft identification by moment invariants. IEEE Trans. Comput. C-26(1): 39-45; 1977.

7. S. Maitra. Moment invariants. Proc. IEEE, 67: 697-699; 1979.

8. E. Kreyszig. Introduction to Differential Geometry and Riemannian Geometry. University of Toronto Press, Toronto, Ont. 1968.

9. P.J. Besl and R. Jain. Invariant surface characteristics for 3D object recognition in range images. Comput. Vision, Graphics, Image Process. 33: 33-80; 1986.

10. W.I. Grosky and R. Jain. Pyramid-based approach to segmentation applied to region matching, IEEE Trans. Pattern Anal. Mach. Intell. PAMI-8: 639-650; 1986.

11. Y.F. Wang, M.J. Magee, and J.K. Aggarwal. Matching three-dimensional objects using silhouettes. IEEE Trans. Pattern Anal. Mach. Intell. PAMI-6: 513-518; 1984.

12. F.L. Alt. Digital pattern recognition by moments. ACM J. 9: 240-258; 1962.

13. M. Rioux. Laser range finder on synchronized scanners. Appl. Opt. 23: 3837-3844; 1984.

14. F. Bumbaca, F. Blais, and M. Rioux. Real-time correction of three-dimensional nonlinearities for a laser rangeefinder. Opt. Eng. 25(4): 561-565; 1986.

15. F.R. Livingstone and M. Rioux. Optical techniques for industrial inspection. Proc. SPIE - The International Society for Optical Engineering, 665; 188-194; 1986.

16. F.B. Seely, N.E. Ensign, and P.G. Jones. Analytical Mechanics for Engineers. 5th Edition. John Wiley, New York, NY. 1958.

APPENDIX

```
C                        "DRSHOW"
C     THIS  PROGRAM  IS  A DRIVER FOR MATCHING '3-D' OBJECTS
C OBTAINED BY A LASER RANGE CAMERA, USING  CERTAIN INVARIANT
C PARAMETERS OF THE OBJECT. THIS DRIVER CALCULATES 13 INVAR-
C IANT MATCHING PARAMETERS.
C     THIS DRIVER CALLS THE  MAIN SUBROUTINE 'DR3DRANGE'.
C
C     FOR THE MEANING OF THE PARAMETERS SEE THE COMMENTS  IN
C THE MAIN SUBROUTINE DR3DRANGE.
C
C     WE HAVE 9 OBJECTS (MODELS), 7 GIVEN  AND  2 SIMULATED.
C EACH IS TRANSLATED  AND ROTATED ABOUT ITS Z-AXIS AN ARBIT-
C RARY ANGLE. EACH OF MODELS 1, 2  AND 8 IS  TRANSLATED  AND
C ROTATED 4 TIMES WHILE THE OTHER MODELS ARE  TRANSLATED AND
C ROTATED 5 TIMES EACH.  THEREFORE  WE HAVE (3X4 + 6X5) = 42
C OBJECT DATA. AN INTEGER K, BETWEEN 1 AND 42 IS FED TO THIS
C DRIVER.   THE OBTAINED RESULT GIVES THE MODEL TO WHICH THE
C OBJECT K BELONGS.
C
C     MATRIX 'DATAMODEL'  CONTAINS  THE  MATCHING PARAMETERS
C FOR THE 9 MODELS.
C
      LOGICAL*1 MAPL(128,128)
      CHARACTER*1 MATSGN(128,128),MATCHAR(128,128),KONTMAP
     *(128,128)
C
      INTEGER KONTXPTS(1024),KONTYPTS(1024)
      DIMENSION DUMPMAP(128,128)
C
      DIMENSION PICTUR(128,128),SPICTUR(128,128),FU(128,
     1128),FV(128,128),FUV(128,128),FUU(128,128),FVV(128,
     2128),V(128),EKEE(128,128),EHEE(128,128),SKEE(128,
     3128),SHEE(128,128),EDGMAP(128,128),FLATMAP(128,128),
     4ERRSURF(128,128),MAP(128,128)
C
      DIMENSION C(16384,7),X(7),A(7),ANEW(7),R(7,7),
     1RES(16384),W(16384),DM(16384),EL(7,7),Y(7),AMMT(7),
     2AMMTKNT(7)
C
      DIMENSION XPTS(16384),YPTS(16384),ZED(16384),
     1PICTURROT(128,128),XROT(16384),YROT(16384)
C
      DIMENSION ERROR(9,25),SHOW(25),IS(29),IC(50),
     1DATAMODEL(9,14),SCALEDAT(9,14),SCALE(14),OBJECTSCL
     2(14),IMATCH(14)
C
      CHARACTER STRING*11
      DATA STRING/'0123456789 '/
C
C    THE SHOWN DATA  'DATAMODEL'  IS  THE  RESULT OF RUNNING
C THE DRIVER DRMATCH. IT IS GIVEN HERE FOR  COMPARISON TESTS
```

```
      DATA DATAMODEL/0.0,1.0,1.0,0.0,0.0,0.0,1.0,0.0,0.0,
     10.57,1.13,1.38,0.90,0.84,0.66,2.25,0.65,0.66,
     252.98,60.91,32.68,84.21,57.49,74.55,31.47,66.80,31.16,
     343.63,32.54,31.17,36.23,30.84,55.89,63.32,50.53,23.42,
     45990.,3266.,2859.,2639.,2769.,3708.,2455.,4222.,1178.,
     5100398.,38487.,32299.,38872.,39564.,63961.,63575.,
     *149504.,13986.0,
     616.76,11.78,11.30,14.73,14.29,17.25,25.90,35.41,11.87,
     711.95,5.01,5.54,7.58,6.53,6.06,8.44,5.55,4.53,
     840.19,32.23,27.46,29.82,31.83,35.78,29.48,36.47,19.54,
     926.15,24.06,25.14,18.88,21.26,25.87,23.24,25.28,13.28,
     138013956.,11177562.,15309412.,7846435.,13319996.,
     *32173478.,28406098.,60419772.,1535426.,
     330614870.,11105398.,5100946.,6012434.,4564126.,
     *10619395.,5937623.,35124372.,932453.,
     3508255.,165993.,198692.,194034.,238066.,303936.,
     *183349.,238124.,37219.,
     4403474.,133482.,130010.,82514.,84531.,126419.,
     *62891.,150414.,24898./
C
  101 FORMAT(40H0  ******START DRIVER  DRSHOW********   )
  102 FORMAT(25H0    IMATCH(I), I=2:14   )
  103 FORMAT(25H0    IS(I)    , I=2:14   )
  104 FORMAT(1X,14I5)
  105 FORMAT(40H0  SCALED DATAMODEL MATRIX             )
  106 FORMAT(40H0  SCALED PARAMETERS OF THE OBJECT     )
  107 FORMAT(1X,13F6.1)
  108 FORMAT(37H0 IMAPL,IMAPLC,IKONT,ISMOOTH,NSMOOTH ,5I5)
  109 FORMAT(30H0 NORMSC, NOISE ,SIGMANOIS    ,2I5,F8.2)
  110 FORMAT(38H0 MODEL FOR WHICH OBJECT BELONG IS NO=,I5)
  111 FORMAT(38H0 ANGLE MAJOR AXIS MAKES WITH I-AXIS =,F8.5)
  112 FORMAT(38H0 ANGLE BASE MJ  AXIS MAKES WITH I-AX=,F8.5)
  113 FORMAT(1X,8E9.2)
  114 FORMAT(1X,2F7.2,2F8.2,2F10.2,2F6.2)
  115 FORMAT(38H0 NUMBER  OF  MATCHING   CHARACTERS  =,I5)
  116 FORMAT(22H0 EXEC. DRSHOW TIME1 =,F15.5,10HSECONDS   )
  117 FORMAT(42H0 INDE,ECC,ASEMIMJ,BSEMIMN,KPTS,VOL,ZM,ZSG)
  118 FORMAT(42H0 SGKNT,SGMMT,MMTMX,MMTMN,MMTMXKN,MMTMNKN )
  119 FORMAT(1X,2F7.2,4F10.0)
  120 FORMAT(40H0 AMMT(1),AMMT(2), ... ,AMMT(6),AMMT(7) )
  121 FORMAT(40H0 OBJECT DATA                          )
  122 FORMAT(40H0 MODEL DATA                           )
  124 FORMAT(1X,F10.2,2F9.5)
  125 FORMAT(40H0     FIRST  CHOICE                    )
  126 FORMAT(40H0     SECOND CHOICE                    )
  127 FORMAT(40H0     THIRD  CHOICE                    )
  128 FORMAT(40H0     ********************************  )
  129 FORMAT(40H0  NO   OTHER   MATCHING  CHOICES      )
  130 FORMAT(40H0  THIS  IS  A  STRANGE  OBJECT        )
  135 FORMAT(40H0  ******EXIT DRIVER  DRSHOW**********  )
C
```

```
      WRITE(3,101)
C INITIALIZATION
      PI = 4.0*ATAN(1.0)
      IROW = 128
      KOL = 128
      MAXROW = 128
      MAXKOL = 128
      MAXRAK = 4*(MAXROW + MAXKOL)
      MAXRK = MAXROW*MAXKOL
      MAXX = 7
      KOLX = 4
      EPS = 1.E-04
      MODELNUM = 9
      NUMOBJCT = 42
C
C NOISE = 0: NO GAUSSIAN NOISE IS ADDED TO OBJECT MATRIX.
      NOISE = 0
C SIGMANOIS = AMOUNT OF STANDARD DEVIATION OF ADDED NOISE.
      SIGMANOIS = 2.0
C ISMOOTH = 0: INPUT MATRIX IS NOT SMOOTHED.
      ISMOOTH = 0
C NSMOOTH = SIZE OF SMOOTHING WINDOW, 3X3,5X5 OR 7X7.
      NSMOOTH = 7
C IMAPL = 0: MAPL MATRIX IS NOT CREATED.
      IMAPL = 1
C IMAPLC = 0: NO CURVATURE MATICES ARE CREATED.
      IMAPLC = 0
C IKONT = 0: CONTOUR MAP IS NOT PRINTED.
      IKONT = 0
C NORMSC = 0: THE 7 CENTRAL MOMENT PARAMETERS ARE
C             NOT INVARIANT UNDER A SCALE CHANGE.
C
       NORMSC = 0
C IDISPLAY = 0: THE USER DOES NOT WISH TO DISPLAY
C               THE OBJECT USING SUBROUTINE DISPLAY.
       IDISPLAY = 0
       WRITE(3,108)IMAPL,IMAPLC,IKONT,ISMOOTH,NSMOOTH
       WRITE(3,109)NORMSC,NOISE,SIGMANOIS
C
      ICHARCTR = 14
      SCALEPAR = 255.0
      DO 3 J=2,ICHARCTR
        BIG = -1.0
        DO 1 I=1,MODELNUM
          VALU = DATAMODEL(I,J)
          IF(VALU.GT.BIG) BIG = VALU
    1     CONTINUE
        SCALE(J) = SCALEPAR/BIG
        DO 2 I=1,MODELNUM
    2     SCALEDAT(I,J) = SCALE(J)*DATAMODEL(I,J)
    3   CONTINUE
```

```
C
C     WRITE(3,105)
C     DO 151 I=1,MODELNUM
C 151   WRITE(3,107)(SCALEDAT(I,J),J=2,ICHARCTR)
      DO 4 I=1,33
        IC(I) = I + 14
    4   CONTINUE
      DO 5 I=34,NUMOBJCT
        IC(I) = I + 17
    5   CONTINUE
      ORIGIN = 0.0
C
C MODEL 1 ,   K= 1:4     , IUNIT= 15:18
C MODEL 2 ,   K= 5:8     , IUNIT= 19:22
C MODEL 3 ,   K= 9:13    , IUNIT= 23,27
C MODEL 4 ,   K=14:18    , IUNIT= 28:32
C MODEL 5 ,   K=19:23    , IUNIT= 33:37
C MODEL 6 ,   K=24:28    , IUNIT= 38:42
C MODEL 7 ,   K=29:33    , IUNIT= 43:47
C MODEL 8 ,   K=34:37    , IUNIT= 52:55
C MODEL 9 ,   K=38:42    , IUNIT= 56:60
C
C CHOOSE IUNIT BY SPECIFYING A VALUE FOR K.
C
      K = 15
      IUNIT = IC(K)
      IF(K.GE.34.AND.K.LE.37) NUM = 2
      IF(K.GE.38.AND.K.LE.42) NUM = 6
      IF(K.EQ.34) THETAROT = 0.0
      IF(K.EQ.35) THETAROT = PI/4.0
      IF(K.EQ.36) THETAROT = PI/2.0
      IF(K.EQ.37) THETAROT = 3.0*PI/4.0
      IF(K.EQ.38) THETAROT = 0.0
      IF(K.EQ.39) THETAROT = PI/4.0
      IF(K.EQ.40) THETAROT = PI/2.0
      IF(K.EQ.41) THETAROT = 3.0*PI/4.0
      IF(K.EQ.42) THETAROT = PI
C
      ORIGIN = 0.0
      TIME1 = SECNDS(0.0)
C
C CALLING THE MAIN SUBROUTINE 'DR3DRANGE'.
C
      CALL SETCLK(ORIGIN)
      CALL       DR3DRANGE(IUNIT,JUNIT,NUM,IROW,KOL,EPS,
     1MAXROW,MAXKOL,MAXRK,MAXRAK,MAXX,KOLX,DUMPMAP,PICTUR,
     2SPICTUR,THETAROT,PICTURROT,KPTS,XPTS,YPTS,ZED,XROT,
     3YROT,IKONT,KONTPTS,KONTXPTS,KONTYPTS,MMTMAXKNT,
     4MMTMINKNT,THETAKNT,SIGMAKNT,FU,FV,FUV,FUU,FVV,V,EKEE,
     5EHEE,SKEE,SHEE,EDGMAP,FLATMAP,ERRSURF,C,X,A,ANEW,INDE,
     6ECCENT,ASEMIMAJ,BSEMIMIN,THETAELL,VOLUM,ZMEAN,ZSIGMA,
```

```
     7IMOMNTMAX,IMOMNTMIN,THETAMMT,SIGMAMMT,NORMSC,AMMT,
     8R,RES,W,DM,EL,Y,MAP,MAPL,IMAPL,IMAPLC,MATSGN,MATCHAR,
     9KONTMAP,ISMOOTH,NSMOOTH,NOISE,SIGMANOIS,IDISPLAY)
C
      CALL RDCLK(TIME,ORIGIN)
      DELTATM = SECNDS(TIME1)
      WRITE(3,116)DELTATM
C
C VECTOR SHOW CONTAINS THE INVARIANT PARAMETERS.
C    OF THE TESTED OBJECT.
C
      SHOW(1) =  FLOAT(INDE)
      SHOW(2) = ECCENT
      SHOW(3) = ASEMIMAJ
      SHOW(4) = BSEMIMIN
      SHOW(5) = FLOAT(KPTS)
      SHOW(6) = VOLUM
      SHOW(7) = ZMEAN
      SHOW(8) = ZSIGMA
      SHOW(9) = SIGMAKNT
      SHOW(10) = SIGMAMMT
      SHOW(11) = FLOAT(IMOMNTMAX)
      SHOW(12) = FLOAT(IMOMNTMIN)
      SHOW(13) = FLOAT(MMTMAXKNT)
      SHOW(14) = FLOAT(MMTMINKNT)
C     SHOW(15) = AMMT(1)
C     SHOW(16) = AMMT(2)
C     SHOW(17) = AMMT(3)
C     SHOW(18) = AMMT(4)
C     SHOW(19) = AMMT(5)
C     SHOW(20) = AMMT(6)
C     SHOW(21) = AMMT(7)
C
      DO 7 I=1,MODELNUM
C       IF(SHOW(1).NE.DATAMODEL(I,1)) GO TO 7
        DO 6 K=2,ICHARCTR
          CON = (SHOW(K)-DATAMODEL(I,K))/DATAMODEL(I,K)
    6      ERROR(I,K) = ABS(CON)
    7   CONTINUE
C
      DO 9 K=1,ICHARCTR
        SMALL = 1.E09
        DO 8 I=1,MODELNUM
C         IF(SHOW(1).NE.DATAMODEL(I,1)) GO TO 8
          IF(ERROR(I,K).GT.SMALL) GO TO 8
          SMALL = ERROR(I,K)
          MATCH = I
    8     CONTINUE
        IMATCH(K) = MATCH
    9   CONTINUE
C
```

```
      DO 10 I=1,MODELNUM
   10   IS(I) = 0
C     WRITE(3,103)
C     WRITE(3,104)(IS(I),I=1,MODELNUM)
C
      DO 11 J=2,ICHARCTR
        I = IMATCH(J)
        IS(I) = IS(I) + 1
   11   CONTINUE
      KMAX = 0
      WRITE(3,102)
      WRITE(3,104)(IMATCH(I),I=2,ICHARCTR)
      WRITE(3,103)
      WRITE(3,104)(IS(I),I=1,MODELNUM)
      MAX = -1
      DO 12 I=1,MODELNUM
        K = IS(I)
        IF(K.LT.MAX) GO TO 12
        MAX = K
        IMAX = I
   12   CONTINUE
      MPAR = 3
      IF(MAX.LT.MPAR) GO TO 97
C
      WRITE(3,128)
      WRITE(3,125)
      WRITE(3,128)
      WRITE(3,110)IMAX
      WRITE(3,115)MAX
      WRITE(3,128)
      WRITE(3,121)
      WRITE(3,117)
      WRITE(3,114)(SHOW(I), I=1,8)
      WRITE(3,118)
      WRITE(3,119)(SHOW(I), I=9,ICHARCTR)
      WRITE(3,111)THETAMMT
      WRITE(3,112)THETAKNT
      WRITE(3,128)
      WRITE(3,122)
      WRITE(3,117)
      WRITE(3,114)(DATAMODEL(IMAX,I), I=1,8)
      WRITE(3,118)
      WRITE(3,119)(DATAMODEL(IMAX,I), I=9,ICHARCTR)
      WRITE(3,128)
C
      IF(MAX.LT.MPAR) GO TO 98
      KMAX = KMAX + MAX
      DO 14 J=2,ICHARCTR
   14   OBJECTSCL(J) = SCALE(J)*SHOW(J)
      WRITE(3,106)
      WRITE(3,107)(OBJECTSCL(J),J=2,ICHARCTR)
```

```
      IF(KMAX.EQ.13) GO TO 98
      IS(IMAX) = 0
      MAX = -1
      DO 13 I=1,MODELNUM
        K = IS(I)
        IF(K.LT.MAX) GO TO 13
        MAX = K
        IMAX = I
   13   CONTINUE
      IF(MAX.LT.MPAR) GO TO 98
      WRITE(3,128)
      WRITE(3,126)
      WRITE(3,128)
      WRITE(3,110)IMAX
      WRITE(3,115)MAX
      WRITE(3,122)
      WRITE(3,117)
      WRITE(3,114)(DATAMODEL(IMAX,I), I=1,8)
      WRITE(3,118)
      WRITE(3,119)(DATAMODEL(IMAX,I), I=9,ICHARCTR)
      IF(MAX.LT.MPAR) GO TO 98
      WRITE(3,128)
C
      KMAX = KMAX + MAX
      IF(KMAX.EQ.13) GO TO 98
      IS(IMAX) = 0
      MAX = -1
      DO 15 I=1,MODELNUM
        K = IS(I)
        IF(K.LT.MAX) GO TO 15
        MAX = K
        IMAX = I
   15   CONTINUE
      IF(MAX.LT.MPAR) GO TO 98
      WRITE(3,128)
      WRITE(3,127)
      WRITE(3,128)
      WRITE(3,110)IMAX
      WRITE(3,115)MAX
      WRITE(3,122)
      WRITE(3,117)
      WRITE(3,114)(DATAMODEL(IMAX,I), I=1,8)
      WRITE(3,118)
      WRITE(3,119)(DATAMODEL(IMAX,I), I=9,ICHARCTR)
      IF(MAX.LT.MPAR) GO TO 98
      WRITE(3,128)
      GO TO 99
   97 WRITE(3,128)
      WRITE(3,130)
      WRITE(3,128)
      GO TO 99
```

```
C
   98 WRITE(3,128)
      WRITE(3,129)
      WRITE(3,128)
   99 CONTINUE
      WRITE(3,135)
      STOP
      END
```

```
C                          "DRMATCH"
C     THIS  PROGRAM  IS  A DRIVER  FOR CALCULATING INVARIANT
C CHARACTERISTICS  OF  RANGE  DATA OBJECTS FOR THE   PURPOSE
C OF  MATCHING THE '3-D' OBJECTS.
C     THIS DRIVER CALLS THE MAIN SUBROUTINE 'DR3DRANGE'.
C     FOR THE MEANING OF THE PARAMETERS SEE THE COMMENTS  IN
C THE SUBROUTINE DR3DRANGE.
C
C     ******************************************************
C
      LOGICAL*1 MAPL(128,128)
      DIMENSION DUMPMAP(128,128)
C
      DIMENSION PICTUR(128,128),SPICTUR(128,128),FU(128,
     1128),FV(128,128),FUV(128,128),FUU(128,128),FVV(128,
     2128),V(128),EKEE(128,128),EHEE(128,128),SKEE(128,
     3128),SHEE(128,128),EDGMAP(128,128),FLATMAP(128,128),
     4ERRSURF(128,128),MAP(128,128)
C
      DIMENSION C(16384,7),X(7),A(7),ANEW(7),R(7,7),
     1RES(16384),W(16384),DM(16384),EL(7,7),Y(7),AMMT(7),
     2AMMTKNT(7)
C
      DIMENSION XPTS(16384),YPTS(16384),ZED(16384),
     *PICTURROT(128,128),XROT(16384),YROT(16384)
C
      DIMENSION RESULT(100,25),IC(50)
C
      INTEGER KONTXPTS(1024),KONTYPTS(1024)
C
      CHARACTER*1 MATSGN(128,128),MATCHAR(128,128),
     1KONTMAP(128,128)
C
      DIMENSION DATAMODEL(9,14)
C
      CHARACTER STRING*11
      DATA STRING/'0123456789 '/
C
C    THE SHOWN DATA  'DATAMODEL'  IS  THE  RESULT OF RUNNING
C THE PRESENT DRIVER. IT IS GIVEN HERE FOR  EASY CHECKING OF
C OF THE RESULTS.
C
      DATA DATAMODEL/0.0,1.0,1.0,0.0,0.0,0.0,1.0,0.0,0.0,
     10.57,1.13,1.38,0.90,0.84,0.66,2.25,0.65,0.66,
     252.98,60.91,32.68,84.21,57.49,74.55,31.47,66.80,31.16,
     343.63,32.54,31.17,36.23,30.84,55.89,63.32,50.53,23.42,
     45990.,3266.,2859.,2639.,2769.,3708.,2455.,4222.,1178.,
     5100398.,38487.,32299.,38872.,39564.,63961.,63575.,
     *149504.,13986.0,
     616.76,11.78,11.30,14.73,14.29,17.25,25.90,35.41,11.87,
     711.95,5.01,5.54,7.58,6.53,6.06,8.44,5.55,4.53,
```

```
     840.19,32.23,27.46,29.82,31.83,35.78,29.48,36.47,19.54,
     926.15,24.06,25.14,18.88,21.26,25.87,23.24,25.28,13.28,
     138013956.,11177562.,15309412.,7846435.,13319996.,
     *32173478.,28406098.,60419772.,1535426.,
     330614870.,11105398.,5100946.,6012434.,4564126.,
     *10619395.,5937623.,35124372.,932453.,
     3508255.,165993.,198692.,194034.,238066.,303936.,
     *183349.,238124.,37219.,
     4403474.,133482.,130010.,82514.,84531.,126419.,
     *62891.,150414.,24898./
C
  101 FORMAT(40H0  ******START DRIVER DRMATCH********   )
  102 FORMAT(40H0  MAXROW,MAXKOL,IROW,KOL,ISMOOTH,EPS = )
  103 FORMAT(30H0  K, IUNIT,                    ,2I6)
  104 FORMAT(30H0  K, IUNIT, THETAROT           ,2I6,F8.4)
  105 FORMAT(30H0  INDE, ECCENT, THETAELL       ,I6,2F8.4)
  106 FORMAT(37H0 IMAPL,IMAPLC,IKONT,ISMOOTH,NSMOOTH ,5I5)
  107 FORMAT(30H0 NORMSC, NOISE ,SIGMANOIS      ,2I5,F8.2)
  113 FORMAT(1X,8E9.2)
  114 FORMAT(1X,4F7.2,2F11.2,2F6.2)
  115 FORMAT(22H0 EXEC. DRMATCH TIME =,F15.5,10HSECONDS   )
  116 FORMAT(40H0 RESULTS FOR THE MATRIX OBJECTROT      )
  117 FORMAT(42H0 INDE,ECC,ASEMIMJ,BSEMIMN,KPTS,VOL,ZM,ZSG)
  118 FORMAT(42H0 SGKNT,SGMMT,MMTMX,MMTMN,MMTMXKN,MMTMNKN )
  119 FORMAT(1X,2F7.2,4F10.0)
  120 FORMAT(42H0 AMMT(1),AMMT(2), ...   ,AMMT(6),AMMT(7) )
  121 FORMAT(40H0  ******EXIT DRIVER DRMATCH**********  )
C
      PI = 4.0*ATAN(1.0)
      IROW = 128
      KOL = 128
      MAXROW = 128
      MAXKOL = 128
      MAXRAK = 4*(MAXROW + MAXKOL)
      MAXRK = MAXROW*MAXKOL
      MAXX = 7
      KOLX = 4
      EPS = 1.E-04
C NOISE = 0: NO GAUSSIAN NOISE ADDED TO OBJECT MATRIX.
      NOISE = 0
C SIGMANOIS = AMOUNT OF STANDARD DEVIATION OF ADDED NOISE.
      SIGMANOIS = 2.0
C ISMOOTH = 0: INPUT MATRIX IS NOT SMOOTHED
      ISMOOTH = 0
C NSMOOTH = SIZE OF SMOOTHING WINDOW, 3X3,5X5 OR 7X7.
      NSMOOTH = 7
C IMAPL = 0: MAPL MATRIX IS NOT CREATED.
      IMAPL = 1
C IMAPLC = 0: SKEE, SHEE, EDGMAP, FLATMAP AND ERRSURF
C             MAPS ARE NOT CREATED.
      IMAPLC = 0
```

```
C IKONT = 0: CONTOUR MAP IS NOT PRINTED.
      IKONT = 0
C NORMSC = 0: THE 7 CENTRAL MOMENT PARAMETERS ARE
C             NOT INVARIANT UNDER A SCALE CHANGE.
      NORMSC = 0
C IDISPLAY = 0: THE USER DOES NOT WISH TO DISPLAY
C               THE OBJECT USING SUBROUTINE DISPLAY.
      IDISPLAY = 0
C
      WRITE(3,106)IMAPL,IMAPLC,IKONT,ISMOOTH,NSMOOTH
      WRITE(3,107)NORMSC,NOISE,SIGMANOIS
C
      DO 1 I=1,33
        IC(I) = I + 14
    1   CONTINUE
      DO 2 I=34,42
        IC(I) = I + 17
    2   CONTINUE
      K1 = 1
      K2 = 42
      DO 10 K=K1,K2
        IUNIT = IC(K)
        IKONT = 0
        IF(IUNIT.EQ.51) IKONT=1
        IF(IUNIT.EQ.55) IKONT=1
        IF(K.GE.34.AND.K.LE.37) NUM = 2
        IF(K.GE.38.AND.K.LE.42) NUM = 6
        IF(K.EQ.34) THETAROT = 0.0
        IF(K.EQ.35) THETAROT = PI/4.0
        IF(K.EQ.36) THETAROT = PI/2.0
        IF(K.EQ.37) THETAROT = 3.0*PI/4.0
        IF(K.EQ.38) THETAROT = 0.0
        IF(K.EQ.39) THETAROT = PI/4.0
        IF(K.EQ.40) THETAROT = PI/2.0
        IF(K.EQ.41) THETAROT = 3.0*PI/4.0
        IF(K.EQ.42) THETAROT = PI
        IF(K.LE.33) WRITE(3,103)K,IUNIT
        IF(K.GT.33) WRITE(3,104)K,IUNIT,THETAROT
        JUNIT = 70
        IF(IUNIT.EQ.15) JUNIT = 71
        IF(IUNIT.EQ.19) JUNIT = 72
        IF(IUNIT.EQ.23) JUNIT = 73
        IF(IUNIT.EQ.28) JUNIT = 74
        IF(IUNIT.EQ.33) JUNIT = 75
        IF(IUNIT.EQ.38) JUNIT = 76
        IF(IUNIT.EQ.43) JUNIT = 77
        IF(IUNIT.EQ.51) JUNIT = 78
        IF(IUNIT.EQ.55) JUNIT = 79
        ORIGIN = 0.0
C
C CALLING THE MAIN SUBROUTINE 'DR3DRANGE'.
```

```
C
      CALL       DR3DRANGE(IUNIT,JUNIT,NUM,IROW,KOL,EPS,
     1MAXROW,MAXKOL,MAXRK,MAXRAK,MAXX,KOLX,DUMPMAP,PICTUR,
     2SPICTUR,THETAROT,PICTURROT,KPTS,XPTS,YPTS,ZED,XROT,
     3YROT,IKONT,KONTPTS,KONTXPTS,KONTYPTS,MMTMAXKNT,
     4MMTMINKNT,THETAKNT,SIGMAKNT,FU,FV,FUV,FUU,FVV,V,EKEE,
     5EHEE,SKEE,SHEE,EDGMAP,FLATMAP,ERRSURF,C,X,A,ANEW,INDE,
     6ECCENT,ASEMIMAJ,BSEMIMIN,THETAELL,VOLUM,ZMEAN,ZSIGMA,
     7IMOMNTMAX,IMOMNTMIN,THETAMMT,SIGMAMMT,NORMSC,AMMT,
     8R,RES,W,DM,EL,Y,MAP,MAPL,IMAPL,IMAPLC,MATSGN,MATCHAR,
     9KONTMAP,ISMOOTH,NSMOOTH,NOISE,SIGMANOIS,IDISPLAY)
C
      RESULT(K,1) =  FLOAT(INDE)
      RESULT(K,2) = ECCENT
      RESULT(K,3) = ASEMIMAJ
      RESULT(K,4) = BSEMIMIN
      RESULT(K,5) = FLOAT(KPTS)
      RESULT(K,6) = VOLUM
      RESULT(K,7) = ZMEAN
      RESULT(K,8) = ZSIGMA
      RESULT(K,9) = SIGMAKNT
      RESULT(K,10) = SIGMAMMT
      RESULT(K,11) = FLOAT(IMOMNTMAX)
      RESULT(K,12) = FLOAT(IMOMNTMIN)
      RESULT(K,13) = FLOAT(MMTMAXKNT)
      RESULT(K,14) = FLOAT(MMTMINKNT)
      RESULT(K,15) = AMMT(1)
      RESULT(K,16) = AMMT(2)
      RESULT(K,17) = AMMT(3)
      RESULT(K,18) = AMMT(4)
      RESULT(K,19) = AMMT(5)
      RESULT(K,20) = AMMT(6)
      RESULT(K,21) = AMMT(7)
C
      CALL RDCLK(TIME,ORIGIN)
      WRITE(3,115)TIME
   10 CONTINUE
C
      WRITE(3,117)
      DO 11 K=K1,K2
   11   WRITE(3,114)(RESULT(K,J),J=1,8)
      WRITE(3,118)
      DO 12 K=K1,K2
   12   WRITE(3,119)(RESULT(K,J),J=9,14)
C     WRITE(3,120)
C     DO 13 K=K1,K2
C  13   WRITE(3,113)(RESULT(K,J),J=15,21)
C
      WRITE(3,121)
      STOP
      END
```

```
      SUBROUTINE DR3DRANGE(IUNIT,JUNIT,NUM,IROW,KOL,EPS,
     1MAXROW,MAXKOL,MAXRK,MAXRAK,MAXX,KOLX,DUMPMAP,PICTUR,
     2SPICTUR,THETAROT,PICTURROT,KPTS,XPTS,YPTS,ZED,XROT,
     3YROT,IKONT,KONTPTS,KONTXPTS,KONTYPTS,MMTMAXKNT,
     4MMTMINKNT,THETAKNT,SIGMAKNT,FU,FV,FUV,FUU,FVV,V,EKEE,
     5EHEE,SKEE,SHEE,EDGMAP,FLATMAP,ERRSURF,C,X,A,ANEW,INDE,
     6ECCENT,ASEMIMAJ,BSEMIMIN,THETAELL,VOLUM,ZMEAN,ZSIGMA,
     7IMOMNTMAX,IMOMNTMIN,THETAMMT,SIGMAMMT,NORMSC,AMMT,
     8R,RES,W,DM,EL,Y,MAP,MAPL,IMAPL,IMAPLC,MATSGN,MATCHAR,
     9KONTMAP,ISMOOTH,NSMOOTH,NOISE,SIGMANOIS,IDISPLAY)
C
C     THIS  PROGRAM  IS  A MAIN SUBROUTINE FOR READING '3-D'
C RANGE DATA,  CALCULATING CERTAIN INVARIANT CHARACTERISTICS
C OF THE RANGE OBJECT SURFACE  AND OTHER PHYSICAL PROPERTIES
C AND USING  THE CALCULATED DATA FOR  MATCHING PURPOSES.
C
C     THE DATA MIGHT BE GIVEN IN THE FORM OF (1) 128X128 I*4
C MATRIX,   (2) 256X256  REAL*4 MATRIX  OR   (3) IN A REAL*4
C STRING FORM.  BY A STRING FORM WE MEAN A VECTOR WHOSE ELE-
C MENTS ARE: X(1), Y(1), Z(1), X(2), Y(2), Z(2), ... , X(K),
C Y(K), Z(K), WHERE K IS THE NUMBER OF SURFACE POINTS OF THE
C OBJECT.  (4) SIMULATED DATA  IN THE FORM OF 128X128 REAL*4
C MATRIX.
C
C     THIS SUBROUTINE  ACCORDINGLY  CALLS ONE OF THE SUBROU-
C TINES: (1) READIMAGE, (2) READIMAG2,  (3) READSTRING   OR
C (4) OBJECT.
C
C     THEN IT CALLS THE FOLLOWING SUBROUTINES AS  NEEDED:
C SMOOTHIMG, BOUNDS, MOMENTAXS, SCALEP, DISPLAY,   EQUATION,
C NORMEQLS,  ELLIPSE,  KONTOURS, MOMENTKNT, CONVOL5, SGNMAP,
C MOMENTS.
C
C     *****SUBROUTINE READIMAGE DOES THE FOLLOWING.
C     IT READS A GIVEN INTEGER FILE OF  DIMENSIONS  128X128
C AND UNIT NUMBER 'IUNIT' AND  OUTPUTS IT IN THE FORM OF  A
C REAL 128X128 MATRIX 'PICTUR'. IT ALSO OBTAINS THE LARGEST
C ELEMENT OF PICTUR(I,J).
C     IF DESIRED BY THE USER, A GAUSSIAN NOISE OF ZERO MEAN
C AND STANDARD DEVIATION SIGMANOIS MAY BE ADDED TO THE NON-
C ZERO  ELEMENTS  OF THE INPUT FILE.  THAT IS TO THE OBJECT
C ITSELF NOT TO THE BACKGROUND.
C
C     *****SUBROUTINE READIMAG2 DOES THE FOLLOWING.
C     IT READS A GIVEN REAL*4  FILE OF  DIMENSIONS  256X256
C AND UNIT NUMBER 'IUNIT' OBTAINED BY THE  RANGE CAMERA AND
C REDUCES IT TO A  REAL 128X128  MATRIX  'PICTUR' BY TAKING
C EVERY OTHER PIXEL EVERY OTHER ROW.  IT ALSO  OBTAINS  THE
C LARGEST ELEMENTS OF THE READ FILE.
C     IF DESIRED BY THE USER, A GAUSSIAN NOISE OF ZERO MEAN
C AND STANDARD DEVIATION 'SIGMANOIS'  MAY  BE  ADDED TO THE
```

```
C INPUT NONZERO ELEMENTS OF THE INPUT FILE.  THAT IS TO THE
C OBJECT ITSELF NOT TO THE BACKGROUND.
C
C     *****SUBROUTINE BOUNDS DOES THE FOLLOWING.
C     GIVEN IS A 3-D OBJECT IN THE FORM OF A MATRIX
C 'SPICTUR' WHOSE Z-COORDINATES ARE THE (I,J) VALUES IN THE
C MATRIX.
C     THIS SUBROUTINE CALCULATES: (1) THE (I,J) COORDINATES
C OF THE CORNERS OF THE RECTANGLE INSIDE WHICH THE NONZERO
C VALUES OF Z ARE ENCLOSED. ONE OBTAINS IMIN, IMAX, JMIN AND
C JMAX. (2) THE I AND J-COORDINATES OF THE NONZERO
C Z-COORDINATES OF THE OBJECT IN THE FORM OF A STRING OF THE
C X-COORDINATES (FLOAT(I)-COORDINATES) AND THE CORRESPONDING
C STRINGS OF THE Y- (FLOAT(J)) AND THE Z-COORDINATES, AND
C (3) THE NUMBER OF SUCH NONZERO Z-COORDINATES. (4) IT ALSO
C CREATES A 'LOGICAL*1' FILE MATRIX 'MAPL' OF THE INPUT FILE
C STORES UNDER THE UNIT NUMBER 'JUNIT' FOR DISPLAY PURPOSES
C ON A DISPLAY SCREEN.
C
C     *****SUBROUTINE READSTRNG DOES THE FOLLOWING.
C     THIS SUBROUTINE READS A GIVEN REAL STRING [X(1),Y(1),
C Z(1), X(2),Y(2),Z(2), ... , X(LAST),Y(LAST),Z(LAST)]. IT
C OUTPUTS 3 VECTORS, FOR THE X-, Y- AND Z- CORRESPONDING
C COORDINATES RESPECTIVELY. IT ALSO CALCULATES THE LARGEST
C VALUES OF THE X, Y AND Z COORDINATES AND THE SMALLEST X
C AND Y COORDINATES OF THE GIVEN IMAGE.
C     IF DESIRED BY THE USER, A GAUSSIAN NOISE OF ZERO MEAN
C AND STANDARD DEVIATION 'SIGMANOIS' MAY BE ADDED TO THE Z-
C VALUES OF THE INPUT FILE.
C
C     *****SUBROUTINE OBJECT DOES THE FOLLOWING.
C        THIS SUBROUTINE DOES THE FOLLOWING.
C     (1) IT CREATES SIMPLE SYMMETRIC GEOMETRICAL OBJECTS
C OF DIFFERENT SIZES; SPHERES, ELLIPSOIDES, PARABOLOIDS OF
C REVOLUTION AND CONES TO BE USED FOR TESTING THE INVARIANT
C PARAMETERS OF THE OBJECT, SUCH AS THE CURVATURE AND THE
C MOMENT PARAMETERS. THE CREATED OBJECT IS GIVEN BY MATRIX
C 'PICTUR' WHOSE (I,J) ELEMENT IS THE Z-COORDINATE OF THE
C OBJECT.
C     IF DESIRED BY THE USER, A GAUSSIAN NOISE OF ZERO MEAN
C AND STANDARD DEVIATION 'SIGMANOIS' MAY BE ADDED TO THE
C CREATED OBJECT ITSELF NOT TO THE BACKGROUND.
C     (2) IT CALCUATES 'ZMAX' THE LARGEST Z-COORDINATE OR
C THE LARGEST PICTUR(I,J) ELEMENTS.  ALSO IT CALCULATES
C IMIN, IMAX, JMIN AND JMAX, INTEGER VALUES WHICH ARE THE
C I AND J COORDINATES OF THE RECTANGLE INSIDE WHICH THE
C OBJECT IS ENCLOSED.
C     (3) IT WRITES THE OBTAINED OBJECT MATRIX PICTUR(I,J)
C WITH OR WITHOUT THE ADDED NOISE IN A STRING FORM.  IT
C OBTAINS THE REAL VECTORS XPTS(K), YPTS(K) AND ZED(K),
C WHICH ARE RESPECTIVELY ARE THE X, Y AND Z COORDINATES
```

```
C OF  THE  NONZERO VALUES OF MATRIX PICTUR(I,J).
C     (4) IT WRITES THE  OBTAINED OBJECT MATRIX PICTUR(I,J)
C IN AN  'I*4'  FORMAT  'MAP' FOR DISPLAY PURPOSES  UNDER A
C GIVEN UNIT NUMBER 'IUNIT'.
C     (5) IT ALSO WRITES  A  'LOGICAL*1'  FILE, FOR DISPLAY
C PURPOSES OF THE  OBJECT WITH OR  WITHOUT THE ADDED NOISE.
C IT IS STORED UNDER THE UNIT NUMBER 'JUNIT'.
C
C     *****SUBROUTINE OBJECTROT DOES THE FOLLOWING.
C     (1) IT INPUTS THE VECTORS  XPTS, TPTS AND ZPTS FOR THE
C NONZERO Z-COORDINATES OF THE OBJECT AND ROTATES THE  COOR-
C DINATE POINTS AN  ANGEL 'THETAROT'  RADIANS  ABOUT THE   Z
C AXIS PASSING THROUGH THE MEAN OF THE X-COORDINATES  AND OF
C THE Y-COORDINATES OF THE OBJECT (XMEAN,YMEAN).  THE OUTPUT
C IS  MATRIX OBJECTROR(I,J).
C     (2) IT WRITES THE OBTAINED MATRIX  PICTURROT(I,J) IN A
C STRING FORM. IT OBTAINS  THE REAL VECTORS XPTS(K), YPTS(K)
C AND ZED(K), WHERE ARE THE X, Y AND  Z  COORDINATES  OF THE
C NONZERO VALUES OF THE  OBJECT MATRIX PICTURROT(I,J).
C     (3) IT  CALCULATES  IMIN, IMAX, JMIN AND JMAX  INTEGER
C VALUES WHICH ARE THE COORDINATES OF THE RECTANGLE   INSIDE
C WHICH THE OBJECT IS ENCLOSED.
C     (4) IT WRITES THE  OBTAINED OBJECT MATRIX  PICTURROT(
C I,J) IN AN 'I*4' FORMAT 'MAP' FOR DISPLAY PURPOSES  UNDER
C A  GIVEN UNIT NUMBER 'IUNIT'.
C     (5) IT ALSO WRITES FOR DISPLAY A 'LOGICAL*1'  FILE OF
C THE ROTATED OBJECT WITH OR WITHOUT THE ADDED NOISE, UNDER
C A UNIT NUMBER 'JUNIT'.
C
C     *****SUBROUTINE SMOOTHIMG DOES THE FOLLOWING.
C     THIS SUBROUTINE READS A REAL MATRIX PICTUR AND OUTPUTS
C A REAL MATRIX 'SPICTUR' WHICH IS THE  SMOOTHED  VERSION OF
C 'PICTUR'.  SMOOTHING IS DONE BY  TAKING  THE  AVERAGE OVER
C A 3X3, 5X5 OR 7X7 WINDOW.   IT ALSO CALCULATES THE LARGEST
C ELEMENT OF THE SMOOTHED MATRIX.
C
C     *****SUBROUTINE DISPLAY DOES THE FOLLOWING.
C     THIS SUBROUTINE  DISPLAYS  A  GIVEN PICTURE  FILE SUCH
C THAT EACH OF ITS ELEMENTS IS ONE OF THE CHARACTERS '012345
C 6789 ' IN AN ASCENDING ORDER.
C
C     *****SUBROUTINE MOMENTAXS DOES THE FOLLOWING.
C     THIS  SUBROUTINE INPUTS A SET OF REAL NONEZER  Z-COOR-
C DINATES OF THE OBJECT FOR EACH (X,). THEN IT OBTAINS THE X
C AND Y OBJECT COORDINATES 'XMEAN' AND 'YMEAN',   CALCULATES
C THE MAXIMUM AND MINIMUM MOMENTS OF INERTIA 'IMOMNTMAX, AND
C 'IMOMNTMIN'  AT  (XMEAN,YMEAN)  AND  THE ANGLE 'THETAMMT'
C WHICH THE AXIS OF MAXIMUM MOMENT OF INERTIA MAKES WITH THE
C I-AXIS. ALSO IT CALCULATES THE PARAMETER 'SIGMAMMT'.
C
C     *****SUBROUTINE EQUATION DOES THE FOLLOWING.
```

```
C     THIS SUBROUTINE CALCULATES THE COEFFICIENT MATRIX   OF
C AN OVERDETERMINED SYSTEM OF LINEAR EQUATIONS.    THE LEAST
C SQUARES SOLUTION OF THIS EQUATION GIVES  THE  COEFFICIENTS
C OF THE EQUATION OF THE  ELLIPTIC  (HYPERBOLIC)  PARABOLOID
C WHICH FITS THE NONZERO  Z-COORDINATES OF THE OBJECT IN THE
C LEAST SQUARES SENSE.
C     THE EQUATION OF THE ELLIPTIC  (HYPERBOLIC)  PARABOLOID
C SURFACE IS
C      Z=A(1)+A(2)*X+A(3)*Y+A(4)*X*X+A(5)*X*Y+A(6)*Y*Y    (A)
C OR
C              Z=A(1)+A(4)*X*X+A(5)*X*Y+A(6)*Y*Y          (B)
C
C     *****SUBROUTINE NORMEQLS DOES THE FOLLOWING.
C     GIVEN IS THE REAL  'KPTS' BY 'KOLX' MATRIX 'A' OF RANK
C KOLX.LE.KPTS AND A REAL 'KPTS' VECTOR 'B' FOR THE EQUATION
C                          A*X = B.
C     THIS SUBROUTINE OBTAINS THE NORMAL EQUATION FOR  A*X=B
C THEN CALCULATES THE LEAST SQUARES SOLUTION X OF THE  OVER-
C DETERMINED SYSTEM A*X=B.  FOR SOLVING THE NORMAL EQUATION,
C MATRIX  [A(TRANSPOSE)*A]  IS  FACTORIZED  BY  THE CHOLESKY
C DECOMPOSITION METHOD.
C
C     *****SUBROUTINE ELLIPSE DOES THE FOLLOWING.
C     GIVEN IS A GENERAL EQUATION OF AN ELLIPSE (HYPERBOLA)
C THIS SUBROUTINE CALCULATES MAJOR AND  THE MINOR SEMI-AXES
C OF THE CONIC AND THE ANGLE, THE MAJOR AXIS MAKES WITH THE
C X-AXIS. THE GIVEN EQUATION OF THE ELLIPSE (HYPERBOLA) IS
C     0=A(1)+A(2)*X+A(3)*Y+A(4)*X*X+A(5)*X*Y+A(6)*Y*Y.
C WE WISH TO TRANSFORM THIS EQUATION TO THE STANDARD FORM
C           0=ANEW(1)+ANEW(4)*X*X+ANEW(6)*Y*Y
C FROM WHICH THE MAJOR AND MINOR SEMI-AXES AND THE ECCENTRI-
C CITY OF THE CONIC ARE OBTAINED.
C
C     *****SUBROUTINE KONTOURS DOES THE FOLLOWING.
C     THIS SUBROUTINE  (1) FINDS OUT THE OUTSIDE  AND INSIDE
C CONTOURS OF A TWO DIMENSIONAL IMAGE USING A SIMPLE METHOD.
C (2) IT WRITES  DOWN THE I AND J-COORDINATES OF THE CONTOUR
C OF THE OBJECT IN THE FORM OF A STRING OF THE I-COORDINATES
C AND THE CORRESPONDING STRING  OF  THE  J-COORDINATES,  AND
C (3) THE  NUMBER OF THE OBTAINED CONTOUR POINTS.
C     THE METHOD OF FINDING A CONTOUR POINT IS AS FOLLOWS.
C THE IMAGE MATRIX IS SCANNED COLUMN WISE. CONSIDER NON-ZERO
C ELEMENTS.   EACH  OF  THE  8 NEIGHBOURING ELEMENTS TO THIS
C NON-ZERO ELEMENT AS A CENTER IS EXAMINED. IF ANY OF THE  8
C ELEMENTS IS ZERO, THEN THE CENTER ELEMENT IS A BOUNDARY OR
C A CONTOUR POINT.
C
C     *****SUBROUTINE MOMENTKNT DOES THE FOLLOWING.
C     THIS SUBROUTINE INPUTS A  SET  OF INTEGER BASE CONTOUR
C POINTS WHOSE NUMBER IS  'KONTPTS' AND  WHOSE I AND J COOR-
C DINATES ARE THE INTEGER ARRAYS KONTXPTS AND KONTYPTS.  THE
```

```
C CONTOUR FORMS A BINARY IMAGE.  THE CONTOUR POINTS HAVE THE
C VALUE 1 AND THE REST OF THE IMAGE ZERO.
C     THE  SUBROUTINE  THEN CALCULATES SOME INVARIANT MOMENT
C PARAMETERS OF THE BASE CONTOUR POINTS  AND  THE  PARAMETER
C SIGMAKNT.
C
C     *****SUBROUTINE CONVOL5 DOES THE FOLLOWING.
C     GIVEN IS A 3-D OBJECT IN THE FORM OF THE  IMAGE MATRIX
C 'SPICTUR' WHOSE (I,J) VALUES ARE THE  Z-COORDINATES OF THE
C OBJECT.
C     THIS SUBROUTINE CALCULATES (1) THE 'SMOOTHED GAUSSIAN'
C 'SKEE(I,J)' AND  (2) THE 'SMOOTHED MEAN' 'SHEE(I,J)'  CUR-
C VATURES  AT EACH  POINT (I,J) OF THE SURFACE SPICTUR(I,J).
C THE SMOOTHED VALUE IS THE AVERAGE OVER A 7 BY 7 WINDOW.
C     IT  ALSO  CALCULATES THE 3 SURFACES:  (3) EDGMAP,  (4)
C FLATMAP AND (5) ERRSURF WHICH ARE RESPECTIVELY INDICATIONS
C OF THE EDGES, THE FLATNESS AND A  MEASURE OF THE ERROR  IN
C THE LEAST SQUARES FIT AT (I,J).
C
C     IF DESIRED BY THE USER, THIS SUBROUTINE MAPS  SURFACES
C (1) SKEE, (2) SHEE, (3) EDGMAP,(4) FLATMAP AND (5) ERRSURF
C ONTO 'LOGICAL*1' FILES UNDER CERTAIN UNIT NUMBERS (94, 95,
C ..., 98 RESPECTIVELY).
C     AT EACH POINT (I,J) OF 'SPICTUR', A  QUADRATIC SURFACE
C OF THE  FORM
C          F(I,J) = A00 + A10*I + A01*J + A11*I*J
C                          + A20*(I*I-2) + A02*(J*J-2).
C IS FIT IN A LEAST SQUARES .
C     ONE CALCULATES THE PARTIAL DERIVATIVES OF SPICTUR(I,J)
C W.R.T. U (I.E OF I) AND V (I,E. OF J).  THESE  ARE DENOTED
C BY FU, FV, FUU, FUV  AND  FVV.  THESE IN FACT ARE OBTAINED
C BY CONVOLVING SPICTUR(U,V) WITH  CERTAIN WEIGHTED WINDOWS.
C SEE: P. R. BEAUDET,  ROTATIONAL INVARIANT IMAGE OPERATORS,
C 4TH  ICPR, NOVEMBER 7-10, 1978, KYOTO, JAPAN, PP. 579-583.
C
C     *****SUBROUTINE SGNMAP DOES THE FOLLOWING.
C     GIVEN ARE THE 'SMOOTHED GAUSSIAN CURVATURE' 'SKEE' AND
C THE  'SMOOTHED MEAN CURVATURE' 'SHEE' AT EACH POINT (I,J),
C THIS SUBROUTINE CALCULATES  THE  '8'  COMBINATIONS  OF THE
C '+ 0 -' GAUSSIAN CURVATURE AND THE '+ 0 -' MEAN CURVATURES
C AND PRINTS  A  MAP  OF STRING CHARACTERS '12345678'    FOR
C THIS COMBINATION.
C
C     *****SUBROUTINE MOMENTS DOES THE FOLLOWING.
C     THIS  SUBROUTINE INPUTS A SET OF  REAL  Z-COORDINATES
C OF THE OBJECT FOR EACH X AND Y COORDINATES. IT OBTAINS THE
C MEAN VALUES OF THE X AND Y OBJECT COORDINATES 'XMEAN'  AND
C 'YMEAN' AND CALCULATES THE 7 NORMALIZED  INVARIANT MOMENTS
C AMMT(I),I=1:7.
C     REFERENCES:
C (1) E. HALL,  COMPUTER  IMAGE  PROCESSING AND RECOGNITION,
```

```
C      ACADEMIC PRESS, 1979, PP. 420-421.
C (2) A. GOSHTASBY,  TEMPLATE  MATCHING IN  ROTATED  IMAGES,
C      IEEE TRANS. PATTERN ANALYSIS AND MACHINE INTELLIGENCE,
C      PAMI-7, 338-344, 1985.
C (3) S. MAITRA, MOMENT INVARIANTS, PROCEEDINGS OF THE IEEE,
C      VOL-67, 1979, PP. 697-699.
C
C
C       MEANING OF THE PARAMETERS.
C IROW    AN INTEGER. IT IS THE NUMBER OF ROWS OF THE READ OR
C         THE CREATED (SIMULATED) OBJECT OR IMAGE.
C KOL     AN INTEGER. IT IS THE NUMBER OF COLUMNS OF READ  OR
C         CREATED (SIMULATED) OBJECT OR IMAGE.
C MAXROW AN INTEGER .GE.IROW.
C MAXKOL AN INTEGER .GE. KOL.
C MAXRAK AN INTEGER = 4*(MAXROW + MAXKOL).
C MAXRK   AN INTEGER = MAXROW*MAXKOL.
C IUNIT   AN INTEGER.  IT IS THE UNIT NUMBER OF THE  I*4 READ
C         OR SIMULATED OBJECT OR IMAGE.
C JUNIT   AN INTEGER. IT IS THE UNIT NUMBER OF THE  LOGICAL*1
C         IMAGE OF THE READ IMAGE, FOR MATRIX 'MAPL'.
C NUM     AN INTEGER, THE NUMBER OF THE SIMULATED OBJECT.
C NOISE   AN INTEGER INDICATOR SPECIFIED BY THE USER. NOISE=0
C         INDICATES  NO  NOISE  TO BE ADDED TO MATRIX PICTUR.
C         NOISE.NE.0,  INDICATES  ADDING GAUSSIAN  NOISE WITH
C         ZERO 'MEAN' AND  STANDARD  DEVIATION 'SIGMANOIS' TO
C         PICTUR.
C SIGMANOIS  A  REAL  PARAMETER SPECIFIED BY THE USER. IT IS
C         THE STANDARD DEVIATION OF THE ADDED GAUSSIAN NOISE.
C PICTUR A REAL MAXROW BY MAXKOL MATRIX. ITS FIRST IROW ROWS
C         AND FIRST KOL COLUMNS CONTAIN THE Z-COORDINATES  OF
C         GIVEN  OBJECT  MATRIX  'PICTUR' WITH OR WITHOUT THE
C         ADDED NOISE.
C SPICTUR  A  REAL  'MAXROW' BY 'MAXKOL'  MATRIX  WHICH IS A
C         COPY OF PICTUR OR IT IS THE SMOOTHED PICTUR.
C PICTURROT A REAL MAXROW BY MAXKOL MATRIX.   ITS FIRST IROW
C         ROWS  AND FIRST KOL COLUMNS ARE  THE  Z-COORDINATES
C         OF THE ROTATED OBJECT 'PICTURROT'.
C THETAROT AN ANGLE IN RADIANS, WITH THE X-AXIS   SPECIFIED
C         BY THE USER FOR WHICH 'OBJECT' IS RTATED ABOUT THE
C         Z-AXIS PASSING THROUGH THE C.G. OF PICTUR.
C NSMOOTH AN INTEGER PRAMETER SPECIFIED BY THE USER:   3, 5,
C         OR 7.  IT IS THE SIZE OF THE SMEARING WINDOW.
C ZMAX    A REAL NUMBER WHICH IS THE LARGEST Z-COORDINATE  OF
C         THE OBJECT.THAT IS THE LARGEST PICTUR(I,J) ELEMENT.
C XMAX    A REAL NUMBER WHICH IS THE LARGEST X-COORDINATE  OF
C         THE FILE.
C XMIN    A REAL NUMBER WHICH IS THE SMALLEST X-COORDINATE OF
C         THE FILE
C YMAX    A REAL NUMBER WHICH IS THE LARGEST Y-COORDINATE  OF
C         THE FILE.
```

```
C YMIN    A REAL NUMBER WHICH IS THE SMALLEST Y-COORDINATE OF
C         THE FILE.
C IMIN, IMAX, JMIN AND JMAX
C         INTEGER VALUES WHICH HAVE THE SAME MEANINGS RESPEC-
C         TIVELY AS XMAX, XMIN, YMAX AND YMIN.
C XMEAN   A REAL NUMBER WHICH IS THE MEAN VALUE OF THE X-COO-
C         RDINATES OF THE OBJECT SURFACE.
C YMEAN   A REAL NUMBER WHICH IS THE MEAN VALUE OF THE Y-COO-
C         RDINATES OF THE OBJECT SURFACE.
C ZMEAN   A REAL NUMBER WHICH IS THE MEAN VALUE OF THE Z-COO-
C         RDINATES OF THE OBJECT SURFACE.
C VOLUM   A REAL NUMBER WHICH IS THE VOLUME OF THE OBJECT.
C MAP     AN INTEGER  I*4  'MAXROW' BY 'MAXKOL'  MATRIX WHOSE
C         FIRST IROW ROWS AND FIRST KOL COLUMNS  CONTAIN  THE
C         OUTPUT  OBJECT MATRIX IN  NON-FORMATED FORMAT UNDER
C         THE UNIT NUMBER IUNIT.
C MAPL    A LOGICAL*1 'MAXROW' BY 'MAXKOL' MATRIX WHOSE FIRST
C         IROW ROWS AND FIRST KOL COLUMNS  CONTAIN THE OUTPUT
C         OBJECT MATRIX IN NON-FORMATED FORMAT UNDER THE UNIT
C         NUMBER JUNIT FOR DISPLAY PURPOSES.
C IMAPL   AN INTEGER INDICATOR SPECIFIED BY THE USER. IMAPL=0
C         INDICATES THAT LOGICAL*1 MATRIX 'MAPL' IS NOT TO BE
C         CREATED. IF IMAPL.NE.0, IT IS CREATED
C KPTS    AN INTEGER.   IT IS THE NUMBER OF NONZERO VALUES IN
C         THE MATRIX OF THE OBJECT.
C XPTS    A REAL VECTOR  OF  LENGTH  MAXRK.  ITS  FIRST  KPTS
C         ELEMENTS ARE THE X-COORDINATES OF THE NONZERO VALU-
C         ES OF THE OBJECT MATRIX PICTUR(I,J).
C YPTS    A REAL VECTOR  OF  LENGTH  MAXRK.  ITS  FIRST  KPTS
C         ELEMENTS ARE THE Y-COORDINATES OF THE NONZERO VALU-
C         ES OF THE OBJECT MATRIX PICTUR(I,J).
C ZED     A REAL VECTOR  OF  LENGTH  MAXRK.  ITS  FIRST  KPTS
C         ELEMENTS ARE THE Z-COORDINATES OF THE NONZERO VALU-
C         ES OF THE OBJECT MATRIX PICTUR(I,J).
C KONTPTS AN INTEGER.  IT IS THE NUMBER OF KONTOUR POINTS OF
C         THE GIVEN IMAGE.
C KONTXPTS AN INTEGER VECTOR  OF  LENGTH  MAXRK.  ITS  FIRST
C         KONTPTS ELEMENTS ARE THE I-COORDINATES OF  THE KON-
C         TOUR POINTS.
C KONTYPTS AN INTEGER VECTOR  OF  LENGTH  MAXRK.  ITS  FIRST
C         KONTPTS ELEMENTS ARE THE J-COORDINATES OF THE  KON-
C         TOUR POINTS WHICH CORRESPOND TO THE KONTXPTS POINTS
C KOLX    AN INTEGER .EQ.NUMBER OF COLUMNS IN THE COEFFICIENT
C         MATRIX.  FOR  EQUATION  (A)  ABOVE,  KOLX=6  AND  FOR
C         EQUATION (B) KOLX=4.
C MAXX    AN INTEGER .GE. (KOLX+1). WE TAKE MAXX=7.
C C       A REAL MAXRK BY MAXX MATRIX. ITS FIRST  'KPTS' ROWS
C         AND FIRST 'KOLX' COLUMNS  CONTAIN  THE  COEFFICIENT
C         MATRIX OF THE APPROXIMATING EQUATION.
C B       A REAL MAXRK VECTOR WHOSE FIRST KPTS CONTAIN VECTOR
C         'B' OF THE GIVEN EQUATION C*X=B.
```

```
C X       A REAL 'MAXX' VECTOR  WHOSE FIRST KOLX ELEMENTS ARE
C         THE  LEAST SQUARES SOLUTION OF THE SYSTEM A*X=B.
C RES     A REAL  MAXRK  VECTOR WHOSE FIRST KPTS ELEMENTS ARE
C         THE  RESIDUAL VECTOR (B-A*X) OF A*X=B.
C RHO     A REAL NUMBER WHICH IS THE L2 NORM OF THE  RESIDUAL
C         VECTOR RES.
C R       A REAL 'MAXX BY MAXX'   MATRIX WHOSE FIRST (KOLX+1)
C         ROWS AND FIRST 'KOLX' COLUMNS CONTAIN  THE  PRODUCT
C         A(TRANSPOSE)*(A|B).
C Y       A REAL MAXX WORKING VECTOR.
C EL      A REAL 'MAXX BY MAXX'  MATRIX.  ITS FIRST KOLX ROWS
C         AND FIRST KOLX COLUMNS CONTAIN THE LOWER TRIANGULAR
C         MATRIX  IN THE  CHOLESKY DECOMPOSITION OF THE FIRST
C         KOLX COLUMNS OF MATRIX R WHICH IS  EL*EL(TRANSPOSE)
C A       A  REAL  VECTOR  OF  LENGTH MAXX.  ITS  FIRST  KOLX
C         ELEMENTS  ARE  THE  COEFFICIENTS OF THE EQUATION OF
C         THE ELLIPSE (HYPERBOLA).
C ANEW    A  REAL  VECTOR  OF  LENGTH MAXX.  ITS  FIRST  KOLX
C         ELEMENTS  ARE  THE  COEFFICIENTS OF THE EQUATION OF
C         A STANDARD ELLIPSE (HYPERBOLA). THAT IS IN OUR CASE
C            0 = ANEW(1) + ANEW(4)*X*X + ANEW(6)*Y*Y
C ASEMIMAJ A REAL NUMBER, THE SEMI MAJOR AXIS OF THE CONIC.
C BSEMIMIN A REAL NUMBER, THE SEMI MINOR AXIS OF THE CONIC.
C THETAELL A REAL NUMBER.  IT IS  THE ANGEL (IN RADIANS) THE
C         MAJOR  AXIS  OF  THE ELLIPSE (HYPERBOLA) MAKES WITH
C         THE X-AXIS.
C ECCENT THE  CALCULATED  ECCENTRICITY OF THE ELLIPSE OR THE
C         HYPERBOLA.
C MATCHAR A CHARACTER*1  MAXROW BY MAXKOL MATRIX WHOSE FIRST
C         IROW ROWS AND FIRST KOL COLUMNS CONTAIN THE  OUTPUT
C         CHARACTER MATRIX.
C NORMSC AN INTEGER SPECIFIED BY THE USER.  IF NORMSC=0, THE
C         7 CENTRAL MOMENT PARAMETERS ARE NOT INVARIANT UNDER
C         A SCALE CHANGE.  IF NORMSC.NE.0, THEY ARE INVARIANT
C         UNDER A SCALE CHANGE.
C AMMT    A REAL VECTOR OF LENGTH 7. ITS ELEMENTS ARE THE  7
C         PARAMETERS WHICH ARE  NORMALIZED  INVARIANT CENTRAL
C         MOMENTS FOR THE GIVEN OBJECT.
C FU      A REAL MAXROW BY MAXKOL MATRIX. ITS FIRST IROW ROWS
C         AND FIRST KOL COLUMNS CONTAIN  THE  FIRST   PARTIAL
C         DERIVATIVE  DSPICTUR(U,V)/DU.
C FV      A REAL MAXROW BY MAXKOL MATRIX. ITS FIRST IROW ROWS
C         AND FIRST KOL COLUMNS CONTAIN  THE  FIRST   PARTIAL
C         DERIVATIVE  DSPICTUR(U,V)/DV.
C FUV     A REAL MAXROW BY MAXKOL MATRIX. ITS FIRST IROW ROWS
C         AND FIRST KOL COLUMNS CONTAIN  THE SECOND   PARTIAL
C         DERIVATIVE  D2SPICTUR(U,V)/DUDV.
C FUU     A REAL MAXROW BY MAXKOL MATRIX. ITS FIRST IROW ROWS
C         AND FIRST KOL COLUMNS CONTAIN  THE SECOND   PARTIAL
C         DERIVATIVE  D2SPICTUR(U,V)/DU2.
C FUV     A REAL MAXROW BY MAXKOL MATRIX. ITS FIRST IROW ROWS
```

```
C         AND FIRST KOL COLUMNS CONTAIN  THE SECOND   PARTIAL
C         DERIVATIVE  D2SPICTUR(U,V)/DV2.
C EKEE    A REAL MAXROW BY MAXKOL MATRIX. ITS FIRST IROW ROWS
C         AND  FIRST  KOL  COLUMNS  CONTAIN THE VALUES OF THE
C         GAUSSIAN CURVATURE AT ELEMENTS (I,J).
C EHEE    A REAL MAXROW BY MAXKOL MATRIX. ITS FIRST IROW ROWS
C         AND  FIRST  KOL  COLUMNS  CONTAIN THE VALUES OF THE
C         SMOOTHED MEAN CURVATURE AT ELEMENTS (I,J).
C ABIGK   MAX |SKEE(I,J)|
C ABIGH   MAX |SHEE(I,J)|
C AMAXK,AMAXH,AMAXEDG,AMAXFLAT,AMAXERRS :
C         MAX SKEE(I,J) ,  MAX SHEE(I,J) ,   MAX EDGMAP(I,J),
C         MAX FLATMAP(I,J)  AND MAX ERRSURF(I,J)  RESPECT.
C AMINK,AMINH,AMINEDG,AMINFLAT,AMINERRS :
C         MIN SKEE(I,J) ,  MIN SHEE(I,J) ,   MIN EDGMAP(I,J),
C         MIN FLATMAP(I,J)  AND MIN ERRSURF(I,J)  RESPECT.
C V       A REAL WORKING VECTOR OF DIMENSION MAXROW.
C
C MATSGN A CHARACTER*1 MAXROW BY MAXKOL MATRIX.  ITS   FIRST
C         IROW ROWS AND FIRST KOL COLUMNS CONTAIN THE 8-LEVEL
C         SIGN MATRIX VALUES AT (X,Y).
C EDGMAP A REAL MAXROW BY MAXKOL MATRIX. ITS FIRST IROW ROWS
C         AND  FIRST KOL  COLUMNS  CONTAIN  THE VALUE SQRT(G)
C         =SQRT(1+FU*FU+FV*FV)  AT  ELEMENT  (I,J). THIS IS A
C         MEASURE OF AN EDGE MAGNITUDE AT (I,J).
C FLATMAP A REAL MAXROW BY MAXKOL MATRIX.  ITS  FIRST   IROW
C         ROWS AND FIRST  KOL  COLUMNS  CONTAIN  THE  VALUE :
C         Q  =  (FUU*FUU+2*FUV*FUV+FVV*FVV) AT ELEMENT (I,J).
C         THIS IS A MEASURE OF SURFACE FLATNESS AT (X,Y).
C ERRSURF  A MEASURE OF THE ERROR  IN THE LEAST SQUARES FIT.
C         = A00 - FUU - FVV - SPICTUR(I,J).  IT IS THE RESULT
C         OF THE CONVOLUTION WITH THE 5 BY 5 WINDOW
C
C                      |-65    10     35    10    -65|
C                 1    | 10    85    110    85     10|
C                ---   | 35    110 -740    110    35|
C                875   | 10    85    110    85     10|
C                      |-65    10     35    10    -65|
C
C         HOWEVER, INSTEAD, WE  HERE CALCULATE ERRSURF AS THE
C         RESULT OF THE UNSHARP MASK (SQUARES). SEE E.L HALL,
C         COMPUTER IMAGE PROCESSING AND RECOGNITION, ACADEMIC
C         PRESS, 1979, PP. 202-204.
C IMAPLC AN INTEGER INDICATOR SPECIFIED BY THE USER.      IF
C         IMAPLC.NE.0, THIS  SUBROUTINE MAPS 'SURFACES' : (1)
C         SKEE, (2) SHEE, (3) EDGMAP, (4)  FLATMAP  AND   (5)
C         ERRSURF  ONTO  'LOGICAL*1' FILES UNDER CERTAIN UNIT
C         NUMBERS (94, 95,...98 RESPECTIVELY). IF IMAPLC.EQ.0
C NSIDE   AN INTEGER, THE SIDE LENGTH  OF THE SQUARE CONVOLU-
C         TION WINDOW. WE TAKE NSIDE=5.
C ZSIGMA A REAL  NUMBER  WHICH  IS THE STANDARD DEVIATION OR
```

```
C         THE SQUARE ROOT OF THE VARIANCE OF THE Z CO-ORDINA-
C         TES OF THE OBJECT.
C SIGMAKNT  A REAL  NUMBER  WHICH  IS THE STANDARD DEVIATION
C         OR THE SQUARE ROOT OF  THE  VARIANCE OF THE X AND Y
C         COORDINATES  OF  THE CONTOUR OF THE BASE SILHOUETTE
C         OF THE OBJECT. SEE TEXT.
C MMTMAXKNT AN INTEGER,   THE  MAXIMUM  MOMENT OF INERTIA OF
C         THE CONTOUR OF THE BASE SILHOUETTE  ABOUT  AN  AXIS
C         THROUGH THE CENTROID OF THE CONTOUR.
C MMTMINKNT AN INTEGER,   THE  MANIMUM  MOMENT OF INERTIA OF
C         THE CONTOUR OF THE BASE SILHOUETTE  ABOUT  AN  AXIS
C         THROUGH THE CENTROID OF THE CONTOUR.
C THETAKNT   A REAL NUMBER.  IT  IS  THE  ANGEL (IN RADIANS)
C         BETWEEN  THE  AXIS OF MAXIMUM MOMENT OF INERTIA  OF
C         THE CONTOUR OF THE BASE SILHOUETTE WITH THE PRESENT
C         I- AXIS.
C SIGMAMMT  A REAL  NUMBER  WHICH  IS THE STANDARD DEVIATION
C         OR THE SQUARE ROOT OF  THE  VARIANCE OF THE X AND Y
C         COORDINATES OF THE OBJECT SURFACE. SEE TEXT
C IMOMNTMAX  AN INTEGER,  THE  MAXIMUM  MOMENT OF INERTIA OF
C         THE OBJECT ABOUT AN AXIS THROUGH (XMEAN,YMEAN).
C IMOMNTMIN  AN INTEGER,  THE  MINIMUM  MOMENT OF INERTIA OF
C         THE OBJECT ABOUT AN AXIS THROUGH (XMEAN,YMEAN).
C THETAMMT   A REAL NUMBER.  IT  IS  THE  ANGEL (IN RADIANS)
C         BETWEEN  THE  AXIS OF MAXIMUM MOMENT OF INERTIA AND
C         THE PRESENT I- AXIS.
C EPS     A SPECIFIED TOLERANCE SUCH THAT A CALCULATED NUMBER
C         Z IS CONSIDERED ZERO IF |Z|<EPS. FOR THE VAX 11/780
C         COMPUTER, EPS IS USUALLY TAKEN 1.E-4 AND 1.E-11 FOR
C         SINGLE AND DOUBLE PRECISION RESPECTIVELY.
C INDD    AN INTEGER INDICATOR. INDD=0 INDICATES  NORMAL EXIT
C         INDD=-1 INDICATES THAT KPTS.LT.KOLX,THIS SUBROUTINE
C         IS NOT DESIGNED FOR UNDERDETERMINED SYSTEMS,    OR,
C         MAXX.LT.(KOLX+1), OR MAXX.GT.MAXRK.  INDD=-2 IS FOR
C         PREMATURE EXIT DUE TO RANK DEFICIENCY OF MATRIX A.
C INDE    AN INTEGER INDICATOR.  INDE=0  INDICATES  THAT  THE
C         GIVEN  EQUATION  IS THAT OF AN ELLIPSE.  INDE=1  OR
C         INDE=2 INDICATES  THAT  THE  GIVEN EQUATION IS OF A
C         HYPERBOLA. IND=-1 INDICATES THAT THE GIVEN EQUATION
C         IS FOR NEITHER AN ELLIPSE NOR FOR  A HYPERBOLA  AND
C         THE SUBROUTINE EXITS PREMATURELY.
C IERRORO AN INTEGER ERROR INDICATOR.
C IERRORC AN INTEGER ERROR INDICATOR.
C         **************************************************
C
      LOGICAL*1 MAPL(MAXROW,MAXKOL)
      DIMENSION DUMPMAP(MAXROW,MAXKOL)
C
      DIMENSION PICTUR(MAXROW,MAXKOL),SPICTUR(MAXROW,MAXKOL)
     1,FU(MAXROW,MAXKOL),FV(MAXROW,MAXKOL),FUV(MAXROW,MAXKOL
     2),FUU(MAXROW,MAXKOL),FVV(MAXROW,MAXKOL),V(MAXROW),EKEE
```

```
     3(MAXROW,MAXKOL),EHEE(MAXROW,MAXKOL),SKEE(MAXROW,
     4MAXKOL),SHEE(MAXROW,MAXKOL),EDGMAP(MAXROW,MAXKOL),
     5FLATMAP(MAXROW,MAXKOL),ERRSURF(MAXROW,MAXKOL),
     6MAP(MAXROW,MAXKOL)
C
      DIMENSION C(MAXRK,MAXX),X(MAXX),A(MAXX),ANEW(MAXX),
     1R(MAXX,MAXX),RES(MAXRK),W(MAXRK),DM(MAXRK),EL(MAXX,
     2MAXX),Y(MAXX),AMMT(MAXX)
C
      DIMENSION PICTURROT(MAXROW,MAXKOL),XPTS(MAXRK),YPTS(
     *MAXRK),ZED(MAXRK),XROT(MAXRK),YROT(MAXRK)
C
      INTEGER KONTXPTS(MAXRAK),KONTYPTS(MAXRAK)
C
      CHARACTER*1 MATSGN(MAXROW,MAXKOL),MATCHAR(MAXROW,
     *MAXKOL),KONTMAP(MAXROW,MAXKOL)
C
      REAL*4 TIME,ORIGIN,ORIGIND
C
      CHARACTER STRING*11
      DATA STRING/'0123456789 '/
C
  101 FORMAT(40HO  ******START DRIVER DR3DRANGE******    )
  128 FORMAT(1X,256A1)
  141 FORMAT(25HO EXEC. READIMAGE TIME  =,F12.4,7HSECONDS)
  142 FORMAT(25HO EXEC. READIMAG2 TIME  =,F12.4,7HSECONDS)
  144 FORMAT(25HO EXEC. BOUNDS    TIME  =,F12.4,7HSECONDS)
  145 FORMAT(25HO EXEC. READSTRNG TIME  =,F12.4,7HSECONDS)
  146 FORMAT(25HO EXEC. OBJECT    TIME  =,F12.4,7HSECONDS)
  147 FORMAT(25HO EXEC. OBJECTROT TIME  =,F12.4,7HSECONDS)
  148 FORMAT(25HO EXEC. SMOOTHIMG TIME  =,F12.4,7HSECONDS)
  149 FORMAT(25HO EXEC. DISPLAY   TIME  =,F12.4,7HSECONDS)
  150 FORMAT(25HO EXEC. MOMENTAXS TIME  =,F12.4,7HSECONDS)
  151 FORMAT(25HO EXEC. EQUATION  TIME  =,F12.4,7HSECONDS)
  152 FORMAT(25HO EXEC. NORMEQLS  TIME  =,F12.4,7HSECONDS)
  153 FORMAT(25HO EXEC. ELLIPSE   TIME  =,F12.4,7HSECONDS)
  154 FORMAT(25HO EXEC. KONTOURS  TIME  =,F12.4,7HSECONDS)
  155 FORMAT(25HO EXEC. MMTAXSKNT TIME  =,F12.4,7HSECONDS)
  156 FORMAT(25HO EXEC. CONVOL5   TIME  =,F12.4,7HSECONDS)
  157 FORMAT(25HO EXEC. SGNMAP    TIME  =,F12.4,7HSECONDS)
  158 FORMAT(25HO EXEC. MOMENTS   TIME  =,F12.4,7HSECONDS)
  159 FORMAT(25HO EXEC. DR3DRANGE TIME  =,F12.4,7HSECONDS)
  161 FORMAT(40HO  ******EXIT DRIVER DR3DRANGE********  )
C
      WRITE(3,101)
      IROW2 = 256
      KOL2 = 256
      MAXROW2 = 256
      MAXKOL2 = 256
      ORIGIND = 0.0
      CALL SETCLK(ORIGIND)
```

```
C
      IF (IUNIT.LE.10) IR=1
      IF (IUNIT.LT.15) IR = 5
      IF (IUNIT.GE.15.AND.IUNIT.LE.47) IR = 2
      IF (IUNIT.GE.51.AND.IUNIT.LE.59) IR = 3
C     IF(IUNIT.GT.59.AND.IUNIT.LE.70) IR = 4
C IR = 4 IS FOR READING OBJECTS OF STRING DATA FORM.
      IF(IUNIT.GT.59) IR = 5
      GO TO (1,2,5,4,99), IR
    1 CONTINUE
      ORIGIN = 0.0
      CALL SETCLK(ORIGIN)
      CALL       READIMAGE(IROW,KOL,MAXROW,MAXKOL,IUNIT,
     1PICTUR,SPICTUR,XSCALE,YSCALE,ZSCALE,ZMAX,NOISE,
     2SIGMANOIS)
      CALL RDCLK(TIME,ORIGIN)
      WRITE(3,141)TIME
      GO TO 3
    2 CONTINUE
      ORIGIN = 0.0
      CALL SETCLK(ORIGIN)
      CALL       READIMAG2(IUNIT,IROW,KOL,MAXROW,MAXKOL,
     1PICTUR,SPICTUR,XSCALE,YSCALE,ZSCALE,ZMAX,NOISE,
     2SIGMANOIS)
      CALL RDCLK(TIME,ORIGIN)
      WRITE(3,142)TIME
C
      XSCALE = 2.0*XSCALE
      YSCALE = 2.0*YSCALE
    3 CONTINUE
      ORIGIN = 0.0
      CALL SETCLK(ORIGIN)
      CALL       BOUNDS(IROW,KOL,MAXROW,MAXKOL,MAXRK,IUNIT,
     1SPICTUR,IMIN,IMAX,JMIN,JMAX,ZMAX,KPTS,XPTS,YPTS,ZED,
     2JUNIT,IMAPL,MAPL)
      CALL RDCLK(TIME,ORIGIN)
      WRITE(3,144)TIME
      GO TO 6
    4 CONTINUE
      ORIGIN = 0.0
      CALL SETCLK(ORIGIN)
      CALL       READSTRNG(KPTS,MAXRK,XPTS,YPTS,ZED,IUNIT,
     *XMAX,XMIN,YMAX,YMIN,ZMAX,NOISE,SIGMANOIS)
      CALL RDCLK(TIME,ORIGIN)
      WRITE(3,145)TIME
      GO TO 8
    5 ORIGIN = 0.0
      CALL SETCLK(ORIGIN)
      CALL       OBJECT(IROW,KOL,MAXROW,MAXKOL,MAXRK,NUM,
     1PICTUR,SPICTUR,IMIN,IMAX,JMIN,JMAX,ZMAX,KPTS,XPTS,
     2YPTS,ZED,IUNIT,MAP,JUNIT,IMAPL,MAPL,NOISE,SIGMANOIS,
```

```
     3IERRORO)
      CALL RDCLK(TIME,ORIGIN)
      WRITE(3,146)TIME
C
      XSCALE = 1.0
      YSCALE = 1.0
      PI = 4.0*ATAN(1.0)
      IF(THETAROT.EQ.0.0) GO TO 6
C
      ORIGIN = 0.0
      CALL SETCLK(ORIGIN)
      CALL        OBJECTROT(IROW,KOL,MAXROW,MAXKOL,MAXRK,
     1THETAROT,PICTURROT,SPICTUR,IMIN,IMAX,JMIN,JMAX,
     2ZMAX,KPTS,XPTS,YPTS,ZED,XROT,YROT,IUNIT,MAP,JUNIT,
     3IMAPL,MAPL)
      CALL RDCLK(TIME,ORIGIN)
      WRITE(3,147)TIME
C
    6 IF(ISMOOTH.EQ.0) GO TO 8
C ISMOOTH.NE.0, THIS PROGRAM DOES SMOOTH THE INPUT IMAGE.
C
      ORIGIN = 0.0
      CALL SETCLK(ORIGIN)
      CALL        SMOOTHIMG(NSMOOTH,IROW,KOL,MAXROW,MAXKOL,
     *IMIN,IMAX,JMIN,JMAX,PICTUR,SPICTUR,ZMAX)
      CALL RDCLK(TIME,ORIGIN)
      WRITE(3,148)TIME
C
    8 CONTINUE
      IF(IDISPLAY.EQ.0) GO TO 9
      ORIGIN = 0.0
      CALL SETCLK(ORIGIN)
      CALL        DISPLAY(IROW,KOL,MAXROW,MAXKOL,MAXRK,
     *KPTS,XPTS,YPTS,ZMAX,ZED,MATCHAR)
      CALL RDCLK(TIME,ORIGIN)
      WRITE(3,149)TIME
C
    9 CONTINUE
      ORIGIN = 0.0
      CALL SETCLK(ORIGIN)
      CALL        MOMENTAXS(KPTS,MAXRK,XPTS,YPTS,ZED,VOLUM,
     1XMEAN,YMEAN,ZMEAN,ZSIGMA,IMOMNTMAX,IMOMNTMIN,SIGMAMMT,
     2THETAMMT,EPS)
      CALL RDCLK(TIME,ORIGIN)
      WRITE(3,150)TIME
C
      ORIGIN = 0.0
      CALL SETCLK(ORIGIN)
      CALL        EQUATION(MAXRK,MAXX,KOLX,KPTS,XMEAN,YMEAN,
     *C,XPTS,YPTS)
      CALL RDCLK(TIME,ORIGIN)
```

```
      WRITE(3,151)TIME
C
      ORIGIN = 0.0
      CALL SETCLK(ORIGIN)
      CALL       NORMEQLS(KPTS,KOLX,MAXRK,MAXX,C,ZED,A,R,
     *EL,Y,EPS,RES,RHO,INDD)
      CALL RDCLK(TIME,ORIGIN)
      WRITE(3,152)TIME
C
      ORIGIN = 0.0
      CALL SETCLK(ORIGIN)
      CALL       ELLIPSE(KOLX,MAXX,A,ANEW,ASEMIMAJ,
     *BSEMIMIN,THETAELL,EPS,ECCENT,INDE)
      CALL RDCLK(TIME,ORIGIN)
      WRITE(3,153)TIME
C
      IF(IR.EQ.4) GO TO 10
C
      ORIGIN = 0.0
      CALL SETCLK(ORIGIN)
      CALL       KONTOURS(IMIN,IMAX,JMIN,JMAX,IROW,KOL,
     1MAXROW,MAXKOL,MAXRAK,SPICTUR,KONTXPTS,KONTYPTS,
     2KONTPTS,MATCHAR,IKONT,EPS)
      CALL RDCLK(TIME,ORIGIN)
      WRITE(3,154)TIME
C
      ORIGIN = 0.0
      CALL SETCLK(ORIGIN)
      CALL       MMTAXSKNT(KONTPTS,MAXRAK,KONTXPTS,KONTYPTS,
     *MMTMAXKNT,MMTMINKNT,THETAKNT,SIGMAKNT)
      CALL RDCLK(TIME,ORIGIN)
      WRITE(3,155)TIME
C
      NSIDE = 5
      ICONVOL = 0
      IF(ICONVOL.EQ.0) GO TO 10
      ORIGIN = 0.0
      CALL SETCLK(ORIGIN)
      CALL       CONVOL5(IROW,KOL,MAXROW,MAXKOL,MAXRAK,
     1NSIDE,SPICTUR,IMIN,IMAX,JMIN,JMAX,XSCALE,YSCALE,
     2FU,FV,FUV,FUU,FVV,EKEE,EHEE,SKEE,SHEE,EDGMAP,
     3FLATMAP,ERRSURF,V,MAPL,IMAPLC,ABIGK,ABIGH,IERRORC)
      CALL RDCLK(TIME,ORIGIN)
      WRITE(3,156)TIME
C
      ORIGIN = 0.0
      CALL SETCLK(ORIGIN)
      CALL       SGNMAP(IROW,KOL,MAXROW,MAXKOL,MAXRK,SKEE,
     1SHEE,MATSGN,KPTS,XPTS,YPTS,ZED)
      CALL RDCLK(TIME,ORIGIN)
      WRITE(3,157)TIME
```

```
C
   10 CONTINUE
C     IMOMENT = 0
C     IF(IMOMENT.EQ.0) GO TO 99
      ORIGIN = 0.0
      CALL SETCLK(ORIGIN)
      CALL       MOMENTS(NORMSC,MAXRK,MAXX,KPTS,XPTS,
     *YPTS,ZED,VOLUM,XMEAN,YMEAN,AMMT,EPS)
      CALL RDCLK(TIME,ORIGIN)
      WRITE(3,158)TIME
C
   99 CONTINUE
      CALL RDCLK(TIME,ORIGIND)
      WRITE(3,159)TIME
      WRITE(3,161)
      RETURN
      END
```

```
      SUBROUTINE READIMAGE(IROW,KOL,MAXROW,MAXKOL,IUNIT,
     1PICTUR,SPICTUR,XSCALE,YSCALE,ZSCALE,ZMAX,NOISE,
     2SIGMANOIS)
C
C     THE SUBROUTINE DOES THE FOLLOWING.
C     IT READS A GIVEN INTEGER FILE OF  DIMENSIONS  128X128
C AND UNIT NUMBER 'IUNIT' AND  OUTPUTS IT IN THE FORM OF  A
C REAL 128X128 MATRIX 'PICTUR'. IT ALSO OBTAINS THE LARGEST
C ELEMENT OF PICTUR(I,J).
C     IF DESIRED BY THE USER, A GAUSSIAN NOISE OF ZERO MEAN
C AND STANDARD DEVIATION SIGMANOIS MAY BE ADDED TO THE NON-
C ZERO  ELEMENTS  OF THE INPUT FILE.  THAT IS TO THE OBJECT
C ITSELF NOT TO THE BACKGROUND.
C
C      INPUT DATA TO SUBROUTINE.
C IROW    AN INTEGER. IT IS THE NUMBER OF ROWS OF PICTUR.
C KOL     AN INTEGER. IT IS THE NUMBER OF COLUMNS OF PICTUR.
C MAXKOL AN INTEGER .GE. KOL.
C MAXROW AN INTEGER .GE.IROW.
C IUNIT   AN INTEGER.  IT IS THE UNIT NUMBER OF THE  I*4 READ
C         IMAGE.
C NOISE   AN INTEGER INDICATOR SPECIFIED BY THE USER. NOISE=0
C         INDICATES  NO  NOISE  TO BE ADDED TO MATRIX PICTUR.
C         NOISE.NE.0,  INDICATES  ADDING GAUSSIAN  NOISE WITH
C         ZERO 'MEAN' AND  STANDARD  DEVIATION 'SIGMANOIS' TO
C         PICTUR.
C SIGMANOIS  A  REAL  PARAMETER SPECIFIED BY THE USER. IT IS
C         THE STANDARD DEVIATION OF THE ADDED GAUSSIAN NOISE.
C
C      OUTPUT OF THE SUBROUTINE.
C PICTUR A REAL 'MAXROW' BY 'MAXKOL' MATRIX WHOSE FIRST IROW
C         ROWS  AND  FIRST  KOL  COLUMNS  CONTAIN  THE OUTPUT
C         MATRIX PICTUR, WITH OR WITHOUT ADDED NOISE.
C SPICTUR  A  REAL  'MAXROW' BY 'MAXKOL'  MATRIX  WHICH IS A
C         COPY OF PICTUR.
C ZMAX    A REAL NUMBER WHICH IS THE LARGEST Z-COORDINATE  OF
C         THE IMAGE. THAT IS THE LARGEST PICTUR(I,J) ELEMENT.
C ZSCALE, XSCALE, YSCALE: REAL NUMBER; Z, X AND Y SCALES FOR
C         THE READ DATA.
C
      DIMENSION LL(128)
      DIMENSION PICTUR(MAXROW,MAXKOL),SPICTUR(MAXROW,MAXKOL)
C
  104 FORMAT(30HO ZSCALE,XSCALE,YSCALE    =    ,3E11.4)
  110 FORMAT(1X,I4,20F4.0)
  111 FORMAT(1X,4X,20F4.0)
  112 FORMAT(40HO  THE UNSCALED EXTRACTED MATRIX PICTURE)
C
C INITIALIZATION.
      AMEAN = 0.0
      IY = 11
```

```
      EPS = 1.0E-04
      ZSCALE = 1.0
      XSCALE = 1.0
      YSCALE = 1.0
      WRITE(3,104)ZSCALE,XSCALE,YSCALE
      ZMAX = -1.0
      REWIND IUNIT
      DO 3 J=1,KOL
        READ(IUNIT) (LL(I),I=1,IROW)
        DO 2 I=1,IROW
          S = LL(I)/ZSCALE
          IF(NOISE.EQ.0) GO TO 1
          IF(S.LT.EPS) GO TO 1
C SUBROUTINE 'NSRAN' IS FOR ADDING GAUSSIAN NOISE
C   OF MEAN 'AMEAM' AND STANDARD DEVIATION SIGMANOIS.
          CALL NSRAN(IY,SIGMANOIS,AMEAN,VE)
          S = S + VE
          IF(S.LT.0.0) S = 0.0
    1     IF(S.GT.ZMAX) ZMAX = S
          PICTUR(I,J) = S
          SPICTUR(I,J) = S
    2     CONTINUE
    3   CONTINUE
      IW = 0
      IF(IW.EQ.0) GO TO 88
      WRITE(3,112)
      DO 87 I=1,IROW
        WRITE(3,110)I,(PICTUR(I,J),J=1,20)
        WRITE(3,111)(PICTUR(I,J),J=21,40)
        WRITE(3,111)(PICTUR(I,J),J=41,60)
        WRITE(3,111)(PICTUR(I,J),J=61,80)
        WRITE(3,111)(PICTUR(I,J),J=81,100)
        WRITE(3,111)(PICTUR(I,J),J=101,120)
        WRITE(3,111)(PICTUR(I,J),J=121,128)
   87   CONTINUE
   88   CONTINUE
      CLOSE(IUNIT)
      RETURN
      END
```

```
      SUBROUTINE READIMAG2(IUNIT,IROW,KOL,MAXROW,MAXKOL,
     1PICTUR,SPICTUR,XSCALE,YSCALE,ZSCALE,ZMAX,NOISE,
     2SIGMANOIS)
C
C     THE SUBROUTINE DOES THE FOLLOWING.
C     IT READS A GIVEN REAL*4  FILE OF  DIMENSIONS  256X256
C AND UNIT NUMBER 'IUNIT' OBTAINED BY THE  RANGE CAMERA AND
C REDUCES IT TO A  REAL 128X128  MATRIX  'PICTUR' BY TAKING
C EVERY OTHER PIXEL EVERY OTHER ROW.  IT ALSO  OBTAINS  THE
C LARGEST ELEMENTS OF THE READ FILE.
C     IF DESIRED BY THE USER, A GAUSSIAN NOISE OF ZERO MEAN
C AND STANDARD DEVIATION 'SIGMANOIS'  MAY  BE  ADDED TO THE
C INPUT NONZERO ELEMENTS OF THE INPUT FILE.  THAT IS TO THE
C OBJECT ITSELF NOT TO THE BACKGROUND.
C
C      INPUT DATA TO SUBROUTINE.
C IROW    AN INTEGER. IT IS THE NUMBER OF ROWS OF PICTUR.
C KOL     AN INTEGER. IT IS THE NUMBER OF COLUMNS OF PICTUR.
C MAXKOL AN INTEGER .GE. KOL.
C MAXROW AN INTEGER .GE.IROW.
C IUNIT   AN INTEGER.  IT IS THE UNIT NUMBER OF THE  I*4 READ
C         IMAGE.
C NOISE   AN INTEGER INDICATOR SPECIFIED BY THE USER. NOISE=0
C         INDICATES  NO  NOISE  TO BE ADDED TO MATRIX PICTUR.
C         NOISE.NE.0,  INDICATES  ADDING GAUSSIAN  NOISE WITH
C         'ZERO'  MEAN  AND STANDARD DEVIATION 'SIGMANOIS' TO
C         PICTUR.
C SIGMANOIS  A  REAL  PARAMETER SPECIFIED BY THE USER. IT IS
C         THE STANDARD DEVIATION OF THE ADDED GAUSSIAN NOISE.
C
C      OUTPUT OF THE SUBROUTINE.
C PICTUR A REAL 'MAXROW' BY 'MAXKOL' MATRIX WHOSE FIRST IROW
C         ROWS  AND  FIRST  KOL  COLUMNS  CONTAIN  THE OUTPUT
C         MATRIX PICTUR.
C SPICTUR A COPY OF MATRIX 'PICTUR'
C ZMAX    A REAL NUMBER WHICH IS THE LARGEST Z-COORDINATE  OF
C         THE IMAGE. I.E. THE LARGEST DUMPMAP2(I,J) ELEMENT.
C ZSCALE, XSCALE, YSCALE: REAL NUMBER; Z, X AND Y SCALES FOR
C         THE READ DATA.
C
      DIMENSION ELL(256),DUMPMAP2(256,256)
      DIMENSION PICTUR(MAXROW,MAXKOL),SPICTUR(MAXROW,MAXKOL)
C
  104 FORMAT(30H0 ZSCALE,XSCALE,YSCALE     =    ,3E11.4)
  110 FORMAT(1X,I4,20F4.0)
  111 FORMAT(1X,4X,20F4.0)
  112 FORMAT(40H0  THE UNSCALED EXTRACTED MATRIX PICTURE)
C
C INITIALIZATION.
      AMEAN = 0.0
      IY = 21
```

```
      EPS = 1.0E-04
      ZMAX = -1.0
      VEMAX = -1.0
C
      REWIND IUNIT
      READ(IUNIT)IX,JY,ZSCALE,KX,NORG,XOF,YOF,GRDNULL,
     *XSCALE,YSCALE,XMIN,XMAX,YMIN,YMAX
      WRITE(3,104)ZSCALE,XSCALE,YSCALE
C
      BACKGRND = 27.0
      IF(IUNIT.GE.23) BACKGRND = 63.0
C
      KOL2 = KOL + KOL
      IROW2 = IROW + IROW
      DO 2 J=2,KOL2
        READ(IUNIT) (ELL(I),I=1,IROW2)
        DO 1 I=1,IROW2
    1     DUMPMAP2(I,J) = ELL(I)/ZSCALE - BACKGRND
    2   CONTINUE
      IF(NOISE.NE.0) GO TO 5
      DO 4 J=2,KOL2,2
        JD2 = J/2
        DO 3 I=2,IROW2,2
          ID2 = I/2
          S = DUMPMAP2(I,J)
          IF(S.LT.EPS) S = 0.0
          IF(S.GT.ZMAX) ZMAX = S
          PICTUR(ID2,JD2) = S
    3     SPICTUR(ID2,JD2) = S
    4   CONTINUE
      GO TO 8
    5 DO 7 J=2,KOL2,2
        JD2 = J/2
        DO 6 I=2,IROW2,2
          ID2 = I/2
C SUBROUTINE 'NSRAN' IS FOR ADDING GAUSSIAN NOISE
C   OF MEAN 'AMEAM' AND STANDARD DEVIATION SIGMANOIS.
          CALL NSRAN(IY,SIGMANOIS,AMEAN,VE)
          AVE = ABS(VE)
          IF(AVE.GT.VEMAX) VEMAX = AVE
          S = DUMPMAP2(I,J) + VE
          IF(S.LT.0.0) S = 0.0
          IF(S.GT.ZMAX) ZMAX = S
          PICTUR(ID2,JD2) = S
    6     SPICTUR(ID2,JD2) = S
    7   CONTINUE
    8 CLOSE(IUNIT)
C
      IW = 0
      IF(IW.EQ.0) GO TO 88
      WRITE(3,112)
```

```
      DO 87 I=1,IROW
        WRITE(3,110)I,(PICTUR(I,J),J=1,20)
        WRITE(3,111)(PICTUR(I,J),J=21,40)
        WRITE(3,111)(PICTUR(I,J),J=41,60)
        WRITE(3,111)(PICTUR(I,J),J=61,80)
        WRITE(3,111)(PICTUR(I,J),J=81,100)
        WRITE(3,111)(PICTUR(I,J),J=101,120)
        WRITE(3,111)(PICTUR(I,J),J=121,128)
   87   CONTINUE
   88   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE BOUNDS(IROW,KOL,MAXROW,MAXKOL,MAXRK,IUNIT,
     1SPICTUR,IMIN,IMAX,JMIN,JMAX,ZMAX,KPTS,XPTS,YPTS,ZED,
     2JUNIT,IMAPL,MAPL)
C
C     GIVEN IS A 3-D OBJECT IN THE FORM OF A MATRIX
C 'SPICTUR' WHOSE Z-COORDINATES ARE THE (I,J) VALUES IN THE
C MATRIX.
C
C     THIS SUBROUTINE CALCULATES: (1) THE (I,J) COORDINATES
C OF THE CORNERS OF THE RECTANGLE INSIDE WHICH THE NONZERO
C VALUES OF Z ARE ENCLOSED. ONE OBTAINS IMIN, IMAX, JMIN AND
C JMAX. (2) THE I AND J-COORDINATES OF THE NONZERO
C Z-COORDINATES OF THE OBJECT IN THE FORM OF A STRING OF THE
C X-COORDINATES (FLOAT(I)-COORDINATES) AND THE CORRESPONDING
C STRINGS OF THE Y- (FLOAT(J)) AND THE Z-COORDINATES, AND
C (3) THE NUMBER OF SUCH NONZERO Z-COORDINATES. (4) IT ALSO
C CREATES A 'LOGICAL*1' FILE MATRIX 'MAPL' OF THE INPUT FILE
C STORES UNDER THE UNIT NUMBER 'JUNIT' FOR DISPLAY PURPOSES
C ON A DISPLAY SCREEN.
C
C      INPUT DATA TO THE SUBROUTINE.
C IROW   AN INTEGER, NUMBER OF ROWS OF THE GIVEN OBJECT
C        MATRIX SPICTUR.
C KOL    AN INTEGER, NUMBER OF COLUMNS OF THE GIVEN OBJECT
C        MATRIX SPICTUR.
C MAXKOL AN INTEGER .GE. KOL.
C MAXROW AN INTEGER .GE.IROW.
C MAXRK  AN INTEGER = MAXROW*MAXKOL.
C SPICTUR  REAL 'MAXROW' BY 'MAXKOL' MATRIX WHOSE FIRST IROW
C        ROWS AND FIRST KOL COLUMNS CONTAIN THE MATRIX
C        OF THE GIVEN OBJECT.
C ZMAX   A REAL PARAMETER, THE LARGEST SPICTUR(I,J) ELEMENT.
C JUNIT  AN INTEGER. IT IS THE UNIT NUMBER OF THE LOGICAL*1
C        IMAGE OF THE READ IMAGE, FOR MATRIX 'MAPL'.
C IMAPL  AN INTEGER INDICATOR SPECIFIED BY THE USER. IMAPL=0
C        INDICATES THAT LOGICAL*1 MATRIX 'MAPL' IS NOT TO BE
C        CREATED. IF IMAPL.NE.0, IT IS CREATED
C
C      OUTPUT OF THE SUBROUTINE.
C IMIN, IMAX, JMIN AND JMAX
C        INTEGER VALUES WHICH ARE THE COORDINATES OF THE
C        RECTANGLE INSIDE WHICH THE OBJECT IS ENCLOSED.
C KPTS   AN INTEGER. IT IS THE NUMBER OF NONZERO ELEMENTS OF
C        MATRIX SPICTUR.
C XPTS   A REAL VECTOR OF LENGTH MAXRK. ITS FIRST KPTS
C        ELEMENTS ARE THE X-COORDINATES OF THE NONZERO
C        ELEMENTS OF SPICTUR(I,J).
C YPTS   A REAL VECTOR OF LENGTH MAXRK. ITS FIRST KPTS
C        ELEMENTS ARE THE Y-COORDINATES OF THE NONZERO
C        ELEMENTS OF SPICTUR(I,J).
C ZED    AN REAL VECTOR OF LENGTH MAXRK. ITS FIRST KPTS
```

```
C          ELEMENTS ARE THE NONZERO ELEMENTS OF SPICTUR(I,J).
C MAPL     A LOGICAL*1 'MAXROW' BY 'MAXKOL' MATRIX.  ITS FIRST
C          IROW ROWS AND FIRST KOL COLUMNS CONTAIN  THE OUTPUT
C          FILE MATRIX MAPL FOR DISPLAY PURPOSES.
C
      DIMENSION SPICTUR(MAXROW,MAXKOL),XPTS(MAXRK),
     *YPTS(MAXRK),ZED(MAXRK)
      LOGICAL*1 MAPL(MAXROW,MAXKOL),LOG(4)
      INTEGER*4 ILOG
      EQUIVALENCE (ILOG,LOG(1))
C
      EPSZ = ZMAX/50.0
      DO 2 J=1,KOL
        JMIN = J
        DO 1 I=1,IROW
          IF(SPICTUR(I,J).LT.EPSZ) GO TO 1
          GO TO 3
    1     CONTINUE
    2   CONTINUE
    3 CONTINUE
      DO 5 I=1,IROW
        IMIN = I
        DO 4 J=1,KOL
          IF(SPICTUR(I,J).LT.EPSZ) GO TO 4
          GO TO 6
    4     CONTINUE
    5   CONTINUE
    6 CONTINUE
      DO 8 J=1,KOL
        KOLMJP1 = KOL -J + 1
        JMAX = KOLMJP1
        DO 7 I=1,IROW
          IF(SPICTUR(I,KOLMJP1).LT.EPSZ) GO TO 7
          GO TO 9
    7     CONTINUE
    8   CONTINUE
    9 CONTINUE
      DO 11 I=1,IROW
        IROWMIP1 = IROW - I + 1
        IMAX = IROWMIP1
        DO 10 J=1,KOL
          IF(SPICTUR(IROWMIP1,J).LT.EPSZ) GO TO 10
          GO TO 12
   10     CONTINUE
   11   CONTINUE
   12 CONTINUE
      DO 14 J=1,KOL
        DO 13 I=1,IROW
          MAPL(I,J) = 0
   13     CONTINUE
   14   CONTINUE
```

```
      KPTS = 0
      IF(IMIN.GE.2) IMIN = IMIN - 1
      IF(JMIN.GE.2) JMIN = JMIN - 1
      IF(IMAX.LT.IROW) IMAX = IMAX + 1
      IF(JMAX.LT.KOL) JMAX = JMAX + 1
      SCALE = ZMAX/255.0
      DO 16 J=JMIN,JMAX
        Y = FLOAT(J)
        DO 15 I=IMIN,IMAX
          X = FLOAT(I)
          S = SPICTUR(I,J)
          IF(S.LT.EPSZ) GO TO 15
          KPTS = KPTS + 1
          XPTS(KPTS) = X
          YPTS(KPTS) = Y
          ZED(KPTS) = S
          ILOG = (S/SCALE)
          MAPL(I,J) = LOG(1)
15        CONTINUE
16      CONTINUE
      IF(IMAPL.EQ.0) GO TO 18
      REWIND JUNIT
      DO 17 J=1,KOL
17      WRITE(JUNIT)(MAPL(I,J),I=1,IROW)
      CLOSE(JUNIT)
18    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE READSTRNG(KPTS,MAXRK,XPTS,YPTS,ZED,IUNIT,
     *XMAX,XMIN,YMAX,YMIN,ZMAX,NOISE,SIGMANOIS)
C
C     THIS SUBROUTINE READS A GIVEN REAL STRING  [X(1),Y(1),
C Z(1), X(2),Y(2),Z(2), ... , X(LAST),Y(LAST),Z(LAST)].   IT
C OUTPUTS  3  VECTORS,  FOR  THE X-, Y- AND Z- CORRESPONDING
C COORDINATES RESPECTIVELY.   IT ALSO CALCULATES THE LARGEST
C VALUES  OF  THE  X, Y AND Z COORDINATES AND THE SMALLEST X
C AND Y COORDINATES OF THE GIVEN IMAGE.
C     IF DESIRED BY THE USER,  A GAUSSIAN NOISE OF ZERO MEAN
C AND STANDARD DEVIATION 'SIGMANOIS' MAY BE ADDED TO THE  Z-
C VALUES OF THE INPUT FILE.
C
C      INPUT DATA TO SUBROUTINE.
C IUNIT  AN INTEGER, THE UNIT NUMBER OF THE IMAGE FILE.
C MAXRK  AN INTEGER = 128*128.
C NOISE  AN INTEGER INDICATOR SPECIFIED BY THE USER. NOISE=0
C        INDICATES  NO NOISE  TO BE ADDED TO THE Z-VALUES OF
C        FILE. NOISE.NE.0, INDICATES  ADDING GAUSSIAN  NOISE
C        WITH ZERO MEAN AND STANDARD  DEVIATION SIGMANOIS TO
C        THE Z VALUES OF THE FILE.
C SIGMANOIS  A  REAL  PARAMETER SPECIFIED BY THE USER. IT IS
C        THE STANDARD DEVIATION OF THE ADDED GAUSSIAN NOISE.
C
C      OUTPUT DATA OF THE SUBROUTINE.
C KPTS   AN INTEGER, THE NUMBER OF PIXELS IN THE GIVEN FILE.
C XPTS   A  REAL MAXRK VECTOR. ITS FIRST 'KPTS' ELEMENTS ARE
C        THE X-COORDINATES OF THE PIXELS OF THE FILE.
C YPTS   A REAL  MAXRK VECTOR. ITS FIRST 'KPTS' ELEMENTS ARE
C        THE Y-COORDINATES OF THE PIXELS OF THE FILE.
C ZED    A REAL MAXRK VECTOR.  ITS FIRST 'KPTS' ELEMENTS ARE
C        THE Z-COORDINATES OF THE PIXELS OF THE FILE.
C XMAX   A REAL NUMBER WHICH IS THE LARGEST X-COORDINATE  OF
C        THE FILE.
C XMIN   A REAL NUMBER WHICH IS THE SMALLEST X-COORDINATE OF
C        THE FILE
C YMAX   A REAL NUMBER WHICH IS THE LARGEST Y-COORDINATE  OF
C        THE FILE.
C YMIN   A REAL NUMBER WHICH IS THE SMALLEST Y-COORDINATE OF
C        THE FILE.
C ZMAX   A REAL NUMBER WHICH IS THE LARGEST Z-COORDINATE  OF
C        THE FILE
C
      DIMENSION XPTS(MAXRK),YPTS(MAXRK),ZED(MAXRK)
C
      XMAX = -1.0
      XMIN = 1.0E09
      YMAX = -1.0
      YMIN = 1.0E09
      ZMAX = -1.0
      AMEAN = 0.0
```

```
      IY = 41
      REWIND IUNIT
      DO 2 I=1,MAXRK
        READ(IUNIT,END=3)(XPTS(I),YPTS(I),ZED(I))
        X = XPTS(I)
        IF(X.GT.XMAX) XMAX = X
        IF(X.LT.XMIN) XMIN = X
        Y = YPTS(I)
        IF(Y.GT.YMAX) YMAX = Y
        IF(Y.LT.YMIN) YMIN = Y
C SUBROUTINE 'NSRAN' IS FOR ADDING GAUSSIAN NOISE
C   OF MEAN 'AMEAM' AND STANDARD DEVIATION SIGMANOIS.
        VE = 0.0
        IF(NOISE.NE.0) CALL NSRAN(IY,SIGMANOIS,AMEAN,VE)
        Z = Z + VE
        IF(Z.LT.0.0) Z = 0.0
        Z = ZED(I)
        IF(Z.GT.ZMAX) ZMAX = Z
    2   CONTINUE
    3 CONTINUE
      CLOSE(IUNIT)
      KPTS = I - 1
      RETURN
      END
```

```
      SUBROUTINE OBJECT(IROW,KOL,MAXROW,MAXKOL,MAXRK,NUM,
     1PICTUR,SPICTUR,IMIN,IMAX,JMIN,JMAX,ZMAX,KPTS,XPTS,
     2YPTS,ZED,IUNIT,MAP,JUNIT,IMAPL,MAPL,NOISE,SIGMANOIS,
     3IERRORO)
C
C         THIS  SUBROUTINE DOES THE FOLLOWING.
C      (1) IT CREATES  SIMPLE  SYMMETRIC GEOMETRICAL OBJECTS
C OF DIFFERENT SIZES; SPHERES, ELLIPSOIDES,  PARABOLOIDS OF
C REVOLUTION AND CONES TO BE USED FOR TESTING THE INVARIANT
C PARAMETERS OF THE OBJECT, SUCH AS THE CURVATURE  AND  THE
C MOMENT  PARAMETERS. THE CREATED OBJECT IS GIVEN BY MATRIX
C 'PICTUR'  WHOSE  (I,J) ELEMENT IS THE Z-COORDINATE OF THE
C OBJECT.
C      IF DESIRED BY THE USER, A GAUSSIAN NOISE OF ZERO MEAN
C AND STANDARD DEVIATION  'SIGMANOIS' MAY  BE  ADDED TO THE
C CREATED OBJECT ITSELF NOT TO THE BACKGROUND.
C      (2) IT CALCUATES 'ZMAX' THE  LARGEST  Z-COORDINATE OR
C THE LARGEST PICTUR(I,J) ELEMENTS.     ALSO  IT  CALCULATES
C IMIN, IMAX, JMIN AND JMAX,  INTEGER  VALUES WHICH ARE THE
C I AND J  COORDINATES OF THE RECTANGLE  INSIDE  WHICH  THE
C OBJECT IS ENCLOSED.
C      (3) IT WRITES THE OBTAINED OBJECT MATRIX  PICTUR(I,J)
C WITH  OR  WITHOUT THE ADDED NOISE IN A STRING FORM.     IT
C OBTAINS  THE  REAL VECTORS XPTS(K),  YPTS(K)  AND ZED(K),
C WHICH ARE RESPECTIVELY ARE  THE  X, Y  AND  Z COORDINATES
C OF  THE  NONZERO VALUES OF MATRIX PICTUR(I,J).
C      (4) IT WRITES THE  OBTAINED OBJECT MATRIX PICTUR(I,J)
C IN AN  'I*4'  FORMAT  'MAP' FOR DISPLAY PURPOSES  UNDER A
C GIVEN UNIT NUMBER 'IUNIT'.
C      (5) IT ALSO WRITES  A  'LOGICAL*1'  FILE, FOR DISPLAY
C PURPOSES OF THE  OBJECT WITH OR  WITHOUT THE ADDED NOISE.
C IT IS STORED UNDER THE UNIT NUMBER 'JUNIT'.
C
C      THE INPUT TO THE SUBROUTINE.
C IROW    AN INTEGER: NUMBER OF ROWS OF THE CREATED OBJECT.
C KOL     AN INTEGER: NUMBER OF COLUMNS OF THE OBJECT.
C MAXROW AN INTEGER . GE. IROW.
C MAXKOL AN INTEGER . GE. KOL.
C MAXRK  AN INTEGER = MAXROW*MAXKOL
C NUM     AN INTEGER.LE.9, THE NUMBER OF THE CREATED OBJECT.
C IUNIT  AN INTEGER,  THE  UNIT  NUMBER  OF  THE I*4 CREATED
C         FILE.
C JUNIT  AN INTEGER. IT IS THE UNIT NUMBER OF THE  LOGICAL*1
C         CREATED FILE.
C NOISE  AN INTEGER INDICATOR SPECIFIED BY THE USER. NOISE=0
C         INDICATES NO NOISE TO BE ADDED TO THE OBJECT MATRIX
C         'NOISE.NE.0', INDICATES ADDING GAUSSIAN  NOISE WITH
C         ZERO MEAN AND STANDARD DEVIATION SIGMANOIS.
C SIGMANOIS  A  REAL  PARAMETER SPECIFIED BY THE USER. IT IS
C         THE STANDARD DEVIATION OF THE ADDED GAUSSIAN NOISE.
C IMAPL  AN INTEGER INDICATOR SPECIFIED BY THE USER. IMAPL=0
```

```
C         INDICATES THAT LOGICAL*1 MATRIX 'MAPL' IS NOT TO BE
C         CREATED. IF IMAPL.NE.0, IT IS CREATED
C
C     OUTPUT OF THE SUBROUTINE.
C PICTUR A REAL MAXROW BY MAXKOL MATRIX. ITS FIRST IROW ROWS
C         AND FIRST KOL COLUMNS CONTAIN THE Z-COORDINATES OF
C         PICTUR.  PICTUR IS SYMMETRIC ABOUT AN AXIS PARALELL
C         TO THE X (I-) OR THE Y (J-) AXIS.
C SPICTUR  A  REAL  'MAXROW' BY 'MAXKOL'  MATRIX  WHICH IS A
C         COPY OF PICTUR.
C KPTS    AN INTEGER. IT IS THE NUMBER OF NONZERO ELEMENTS OF
C         MATRIX PICTUR.
C XPTS    A REAL VECTOR  OF  LENGTH  MAXRK.  ITS  FIRST  KPTS
C         ELEMENTS  ARE  THE  X-COORDINATES  OF  THE  NONZERO
C         ELEMENTS OF PICTUR(I,J).
C YPTS    A REAL  VECTOR  OF  LENGTH  MAXRK.  ITS FIRST  KPTS
C         ELEMENTS  ARE  THE  Y-COORDINATES  OF  THE  NONZERO
C         ELEMENTS OF PICTUR(I,J).
C ZED     AN REAL  VECTOR  OF  LENGTH MAXRK.  ITS FIRST  KPTS
C         ELEMENTS ARE THE NONZERO ELEMENTS OF PICTUR(I,J).
C ZMAX    A REAL NUMBER WHICH IS THE LARGEST Z-COORDINATE  OF
C         THE OBJECT OR THE LARGEST PICTUR(I,J) ELEMENT.
C IMIN, IMAX, JMIN AND JMAX
C         INTEGER  VALUES  WHICH  ARE  THE COORDINATES OF THE
C         RECTANGLE INSIDE WHICH THE OBJECT IS ENCLOSED.
C MAP     AN INTEGER  I*4  'MAXROW' BY 'MAXKOL'  MATRIX WHOSE
C         FIRST IROW ROWS AND FIRST KOL COLUMNS  CONTAIN  THE
C         OUTPUT  OBJECT MATRIX IN  NON-FORMATED FORMAT UNDER
C         THE UNIT NUMBER IUNIT.
C MAPL    A LOGICAL*1 'MAXROW' BY 'MAXKOL' MATRIX WHOSE FIRST
C         IROW ROWS AND FIRST KOL COLUMNS  CONTAIN THE OUTPUT
C         OBJECT MATRIX IN NON-FORMATED FORMAT UNDER THE UNIT
C         NUMBER JUNIT.
C IERRORO AN INTEGER ERROR INDICATOR.
C
      DIMENSION PICTUR(MAXROW,MAXKOL),SPICTUR(MAXROW,MAXKOL)
     1,MAP(MAXROW,MAXKOL),XPTS(MAXRK),YPTS(MAXRK),ZED(MAXRK)
      LOGICAL*1 MAPL(MAXROW,MAXKOL),LOG(4)
      INTEGER*4 ILOG
      EQUIVALENCE (ILOG,LOG(1))
C
  113 FORMAT(36H0IROW EXCEDES MAXROW:IERRORO,IROW = ,I3,I8)
  114 FORMAT(36H0 KOL EXCEDES MAXKOL:IERRORO,KOL  = ,I3,I8)
  130 FORMAT(1X,I4,20F4.0)
  141 FORMAT(1X,4X,20F4.0)
  132 FORMAT(40H0  THE UNSCALED EXTRACTED MATRIX PICTURE)
C
C INITIALIZATION.
      IF(IROW.GT.MAXROW) GO TO 90
      IF(KOL.GT.MAXKOL) GO TO 91
      IF(NUM.GT.9) GO TO 92
```

```
      PI = 4.0*ATAN(1.0)
      AMEAN = 0.0
      IY = 31
      ZMAX = -1.0
C
      DO 2 J=1,KOL
        DO 1 I=1,IROW
          SPICTUR(I,J) = 0.0
          MAP(I,J) = 0
          MAPL(I,J) = 0
    1     PICTUR(I,J) = 0.0
    2   CONTINUE
      GO TO (3,5,6,7,8,12,30,33,36,99), NUM
C CREATING SPHERES.
    3 XC = IROW/2
      YC = KOL/2
      ZC = 0.0
      RX = IROW/4
      RY = RX
      RZ = RX
      WRITE(3,131)
  131 FORMAT(40H0 A SPHERE                                    )
      GO TO 9
C CREATING ELLIPSOIDS.
    5 XC = IROW/2 + 5.5
      YC = KOL/2 - 10.25
      ZC = 20.75
      RX = IROW/4.0
      RY = RX + 10.0
      RZ = RX - 10.0
      WRITE(3,133)
  133 FORMAT(40H0 AN ELLIPSOID WITH  RX.NE.RY.NE.RZ           )
      GO TO 9
    6 XC = IROW/2 + 5.5
      YC = KOL/2 - 10.25
      ZC = 20.75
      RX = IROW/4
      RY = RX
      RZ = RX - 10.0
      WRITE(3,134)
  134 FORMAT(40H0 AN ELLIPSOID WITH  RY=RX                    )
      GO TO 9
    7 XC = IROW/2 + 5.5
      YC = KOL/2 - 10.25
      ZC = 20.75
      RX = IROW/4
      RY = RX + 10.0
      RZ = RX
      WRITE(3,135)
  135 FORMAT(40H0 AN ELLIPSOID WITH  RZ=RX                    )
      GO TO 9
```

```
    8 XC = IROW/2 + 5.5
      YC = KOL/2 - 10.25
      ZC = 20.75
      RX = IROW/4
      RY = RX + 10.0
      RZ = RY
    9 RX2 = RX*RX
      RY2 = RY*RY
      X1 = XC - RX
      X2 = XC + RX
      Y1 = YC - RY
      Y2 = YC + RY
      ISTART = X1
      IEND = X2
      JSTART = Y1
      JEND = Y2
      DO 11 J=JSTART,JEND
        Y = J
        YCON2 = (Y - YC)*(Y - YC)/RY2
        DO 10 I=ISTART,IEND
          X = I
          XCON2 = (X - XC)*(X - XC)/RX2
          CON = 1.0 - XCON2 - YCON2
          IF(CON.LT.0.0) GO TO 10
          ZVALUE = ZC + RZ*(CON)**0.5
          IF(ZVALUE.GT.ZMAX) ZMAX = ZVALUE
          PICTUR(I,J) = ZVALUE
   10     CONTINUE
   11   CONTINUE
      GO TO 40
C AN ELLIPTIC PARABOLOID ABOUT THE Y AXIS.
C  (X-XC)**2/A**2 + (Z-ZC)**2/B**2 = (Y-YC).
   12 RX = 3.0
      RZ = 1.5
      WRITE(3,137)
  137 FORMAT(40HO ELLIPTIC PARABOLOID ABOUT THE Y AXIS  )
      RZ = RX
      XC = 42.0
      ZC = 0.0
      YC = XC
      X1 = 25.0
      X2 = 69.0
      Y1 = 42.0
      Y2 = 116.0
      ISTART = X1
      IEND = X2
      JSTART = Y1
      JEND = Y2
      IXC = XC
      IYC = YC + 2.0*(Y2 - Y1)/3.0
      XCEN = IXC
```

```
      YCEN = IYC
      RX2 = RX*RX
      DO 20 J=JSTART,JEND
        YJ = J
        DO 19 I=ISTART,IEND
          IF(YJ.LT.YC) GO TO 19
          XI = I
          XCON2 = (XI - XC)*(XI - XC)/RX2
          CON = (YJ - YC) - XCON2
          IF(CON.LT.0.0) GO TO 19
          ZVALUE = ZC + RZ*(CON)**0.5
          IF(ZVALUE.GT.ZMAX) ZMAX = ZVALUE
          PICTUR(I,J) = ZVALUE
   19     CONTINUE
   20   CONTINUE
      WRITE(3,137)
      GO TO 40
C (X-XC)**2/A**2 + (Z-ZC)**2/B**2 = Y.
   30 XC = 20.0
      ZC = 10.0
      X1 = XC
      X2 = 60.0
      Y1 = ZC
      Y2 = 3.0*YC
      ISTART = X1
      IEND = X2
      JSTART = Y1
      JEND = Y2
      RY = 1.5
      RZ = RY - 1.0
      RY2 = RY*RY
      DO 32 J=JSTART,JEND
        Y = J
        YCON2 = (Y - YC)*(Y - YC)/RY2
        DO 31 I=ISTART,IEND
          PICTUR(I,J) = 0.0
          X = I
          CON = (X - XC) - YCON2
          IF(CON.LT.0.0) GO TO 31
          ZVALUE = ZC + RZ*(CON)**0.5
          IF(ZVALUE.GT.ZMAX) ZMAX = ZVALUE
          PICTUR(I,J) = ZVALUE
   31     CONTINUE
   32   CONTINUE
      WRITE(3,137)
      GO TO 40
C A PARABOLOID OF REVOLUTION ABOUT THE Y AXIS:
C     (X-XC)**2/A**2 + (Z-ZC)**2/A**2 = Y.
   33 XC = IROW/4 - 12.0
      ZC = KOL/4 - 12.0
      X1 = XC
```

```
      X2 = XC + XC
      Y1 = ZC
      Y2 = 40.0
      ISTART = X1
      IEND = X2
      JSTART = Y1
      JEND = Y2
      RX = IROW/4
      RZ = RX
      RX2 = RX*RX
      DO 35 J=JSTART,JEND
        Y = J
        DO 34 I=ISTART,IEND
          PICTUR(I,J) = 0.0
          X = I
          XCON2 = (X - XC)*(X - XC)/RX2
          CON = Y - XCON2
          IF(CON.LT.0.0) GO TO 34
          ZVALUE = ZC + RZ*(CON)**0.5
          IF(ZVALUE.GT.ZMAX) ZMAX = ZVALUE
          PICTUR(I,J) = ZVALUE
   34     CONTINUE
   35   CONTINUE
      WRITE(3,138)
  138 FORMAT(40HO PARABOLOID OF REVOLUTION ABOUT Y AXIS )
      GO TO 40
C CREATING A CONE SURFACE ABOUT THE X AXIS.
   36 XC = IROW/4
      YC = KOL/4
      ZC = 20.75
      ALFA = PI/12.0
      TANA2 = TAN(ALFA)*TAN(ALFA)
      X1 = XC
      X2 = XC + IROW/4.0
      Y1 = YC
      Y2 = YC + KOL/4.0
      ISTART = X1
      IEND = X2
      JSTART = Y1
      JEND = Y2
      DO 38 J=JSTART,JEND
        Y = J
        YCON2 = (Y - YC)*(Y - YC)
        DO 37 I=ISTART,IEND
          X = I
          XCON2 = (X - XC)*(X - XC)*TANA2
          CON = XCON2 - YCON2
          IF(CON.LT.0.0) GO TO 37
          ZVALUE = ZC + (CON)**0.5
          IF(ZVALUE.GT.ZMAX) ZMAX = ZVALUE
          PICTUR(I,J) = ZVALUE
```

```
   37      CONTINUE
   38    CONTINUE
         GO TO 40
   40  CONTINUE
         IMIN = ISTART
         IMAX = IEND
         JMIN = JSTART
         JMAX = JEND
         EPSZ = ZMAX/100.0
         KPTS = 0
         SCALE = ZMAX/220.0
         DO 42 J=JMIN,JMAX
           DO 41 I=IMIN,IMAX
             S = PICTUR(I,J)
             IF(S.LT.EPSZ) GO TO 41
             KPTS = KPTS + 1
             XPTS(KPTS) = I
             YPTS(KPTS) = J
             VE = 0.0
C SUBROUTINE 'NSRAN' IS FOR ADDING GAUSSIAN NOISE
C   OF MEAN 'AMEAM' AND STANDARD DEVIATION SIGMANOIS.
             IF(NOISE.NE.0) CALL NSRAN(IY,SIGMANOIS,AMEAN,VE)
             S = S + VE
             IF(S.LT.0.0) S = 0.0
             ZED(KPTS) = S
             SPICTUR(I,J) = S
             MAP(I,J) = S
             ILOG = (S/SCALE)
             MAPL(I,J) = LOG(1)
   41        CONTINUE
   42      CONTINUE
C
         REWIND IUNIT
         DO 48 J=1,KOL
   48      WRITE(IUNIT)(MAP(I,J),I=1,IROW)
         CLOSE(IUNIT)
C
         IF(IMAPL.EQ.0) GO TO 50
         REWIND JUNIT
         DO 49 J=1,KOL
   49      WRITE(JUNIT)(MAPL(I,J),I=1,IROW)
         CLOSE(JUNIT)
   50  CONTINUE
         IW = 0
         IF(IW.EQ.0) GO TO 88
         WRITE(3,132)
         DO 87 I=1,IROW
           WRITE(3,130)I,(PICTUR(I,J),J=1,20)
           WRITE(3,141)(PICTUR(I,J),J=21,40)
           WRITE(3,141)(PICTUR(I,J),J=41,60)
           WRITE(3,141)(PICTUR(I,J),J=61,80)
```

```
      WRITE(3,141)(PICTUR(I,J),J=81,100)
      WRITE(3,141)(PICTUR(I,J),J=101,120)
      WRITE(3,141)(PICTUR(I,J),J=121,128)
87    CONTINUE
88    CONTINUE
      GO TO 99
90    IERRORO = -1
      WRITE(3,113)IERRORO,IROW
      GO TO 99
91    IERRORO = -2
      WRITE(3,114)IERRORO,KOL
      GO TO 99
92    IERRORO = -3
      GO TO 99
99    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE OBJECTROT(IROW,KOL,MAXROW,MAXKOL,MAXRK,
     1THETAROT,PICTURROT,SPICTUR,IMIN,IMAX,JMIN,JMAX,
     2ZMAX,KPTS,XPTS,YPTS,ZED,XROT,YROT,IUNIT,MAP,JUNIT,
     3IMAPL,MAPL)
C
C     THIS SUBROUTINE DOES THE FOLLOWING.
C     (1) IT INPUTS THE VECTORS  XPTS, TPTS AND ZPTS FOR THE
C NONZERO Z-COORDINATES OF THE OBJECT AND ROTATES THE  COOR-
C DINATE POINTS AN  ANGEL 'THETAROT'  RADIANS  ABOUT THE   Z
C AXIS PASSING THROUGH THE MEAN OF THE X-COORDINATES  AND OF
C THE Y-COORDINATES OF THE OBJECT (XMEAN,YMEAN).  THE OUTPUT
C IS  MATRIX OBJECTROR(1,J).
C     (2) IT WRITES THE OBTAINED MATRIX  PICTURROT(I,J) IN A
C STRING FORM. IT OBTAINS  THE REAL VECTORS XPTS(K), YPTS(K)
C AND ZED(K), WHERE ARE THE X, Y AND  Z  COORDINATES  OF THE
C NONZERO VALUES OF THE  OBJECT MATRIX PICTURROT(I,J).
C     (3) IT  CALCULATES  IMIN, IMAX, JMIN AND JMAX  INTEGER
C VALUES WHICH ARE THE COORDINATES OF THE RECTANGLE   INSIDE
C WHICH THE OBJECT IS ENCLOSED.
C     (4) IT WRITES THE  OBTAINED OBJECT MATRIX  PICTURROT(
C I,J) IN AN 'I*4' FORMAT 'MAP' FOR DISPLAY PURPOSES  UNDER
C A  GIVEN UNIT NUMBER 'IUNIT'.
C     (5) IT ALSO WRITES FOR DISPLAY A 'LOGICAL*1'  FILE OF
C THE ROTATED OBJECT WITH OR WITHOUT THE ADDED NOISE, UNDER
C A UNIT NUMBER 'JUNIT'.
C
C     INPUT TO THE SUBROUTINE.
C IROW    AN INTEGER: NUMBER OF ROWS OF MATRIX PICTURROT.
C KOL     AN INTEGER: NUMBER OF COLUMNS OF MATRIX PICTURROT.
C MAXROW AN INTEGER . GE. IROW.
C MAXKOL AN INTEGER . GE. KOL.
C MAXRK   AN INTEGER = MAXROW*MAXKOL
C KPTS    AN INTEGER. IT IS THE NUMBER OF NONZERO ELEMENTS OF
C         MATRIX PICTUR.
C XPTS    A REAL VECTOR  OF  LENGTH  MAXRK.  ITS  FIRST  KPTS
C         ELEMENTS  ARE  THE  X-COORDINATES  OF  THE  NONZERO
C         ELEMENTS OF PICTUR(I,J).
C YPTS    A REAL  VECTOR  OF  LENGTH  MAXRK.  ITS FIRST  KPTS
C         ELEMENTS  ARE  THE  Y-COORDINATES  OF  THE  NONZERO
C         ELEMENTS OF PICTUR(I,J).
C ZED     AN REAL  VECTOR  OF  LENGTH MAXRK.  ITS FIRST  KPTS
C         ELEMENTS ARE THE NONZERO ELEMENTS OF PICTUR(I,J).
C ZMAX    A REAL PARAMETER, THE LARGEST ELEMENTS OF ZED(K).
C THETAROT  AN ANGLE IN RADIANS, WITH THE X-AXIS   SPECIFIED
C         BY  THE USER FOR WHICH THE OBJECT   IS RTATED ABOUT
C         THE Z-AXIS PASSING THROUGH THE C.G. OF THE OBJECT.
C IUNIT   AN INTEGER,  THE  UNIT  NUMBER  OF  THE I*4 CREATED
C         IMAGE MAP. SEE BELOW
C JUNIT   AN INTEGER. IT IS THE UNIT NUMBER OF THE  LOGICAL*1
C         CREATED IMAGE MAPL. SEE BELOW.
C IMAPL   AN INTEGER INDICATOR SPECIFIED BY THE USER. IMAPL=0
```

```
C        INDICATES THAT LOGICAL*1 MATRIX 'MAPL' IS NOT TO BE
C        CREATED. IF IMAPL.NE.0, IT IS CREATED
C
C     OUTPUT OF THE SUBROUTINE.
C XPTS   A REAL VECTOR OF LENGTH MAXRK. ITS FIRST KPTS ELE-
C        MENTS CONTAIN THE X-COORDINATES OF THE NONZERO
C        VALUES OF MATRIX OBJECTROT.
C YPTS   A REAL VECTOR OF LENGTH MAXRK. ITS FIRST KPTS ELE-
C        MENTS CONTAIN THE Y-COORDINATES OF THE NONZERO
C        VALUES OF MATRIX OBJECTROT.
C PICTURROT A REAL MAXROW BY MAXKOL MATRIX.  ITS FIRST IROW
C        ROWS AND FIRST KOL COLUMNS ARE THE Z-COORDINATES
C        OF THE ROTATED OBJECT 'PICTURROT'.
C SPICTUR A COPY OF PICTURROT.
C IMIN, IMAX, JMIN AND JMAX
C        INTEGER VALUES WHICH ARE THE COORDINATES OF THE
C        RECTANGLE INSIDE WHICH THE ROTATED OBJECT IS
C        ENCLOSED.
C MAP    AN INTEGER REAL 'MAXROW' BY 'MAXKOL' MATRIX WHOSE
C        FIRST IROW ROWS AND FIRST KOL COLUMNS CONTAIN THE
C        OBJECT 'PICTURROT' MATRIX IN NON-FORMATED FORMAT
C        UNDER THE UNIT NUMBER IUNIT.
C MAPL   A LOGICAL*1 'MAXROW' BY 'MAXKOL' MATRIX WHOSE FIRST
C        IROW ROWS AND FIRST KOL COLUMNS CONTAIN THE OUTPUT
C        OBJECT MATRIX IN NON-FORMATED FORMAT UNDER THE UNIT
C        NUMBER JUNIT.
C
C     MEANING OF OTHER DIMENSIONED PARAMETERS.
C XROT AND YROT ARE TWO REAL MAXRK WORKING VECTORS.
C
      DIMENSION PICTURROT(MAXROW,MAXKOL),SPICTUR(MAXROW,MAXK
     1OL),XPTS(MAXRK),YPTS(MAXRK),ZED(MAXRK),XROT(MAXRK),
     2YROT(MAXRK),MAP(MAXROW,MAXKOL)
      LOGICAL*1 MAPL(MAXROW,MAXKOL),LOG(4)
      INTEGER*4 ILOG
      EQUIVALENCE (ILOG,LOG(1))
C
  110 FORMAT(1X,I4,20F4.0)
  111 FORMAT(1X,4X,20F4.0)
  112 FORMAT(40H0  THE UNSCALED EXTRACTED MATRIX PICTURE)
C
      XMIN = 1000
      YMIN = 1000
      XMAX = - 1000
      YMAX = - 1000
      DO 2 J=1,KOL
        DO 1 I=1,IROW
          PICTURROT(I,J) = 0.0
          SPICTUR(I,J) = 0.0
          MAPL(I,J) = 0
    1     MAP(I,J) = 0.0
```

```
    2   CONTINUE
        SUMX = 0.0
        SUMY = 0.0
        DO 3 K=1,KPTS
          SUMX = SUMX + XPTS(K)
    3     SUMY = SUMY + YPTS(K)
        XMEAN = SUMX/KPTS
        YMEAN = SUMY/KPTS
C
        SCALE = ZMAX/220.0
        DO 4 K=1,KPTS
          X = XPTS(K)
          Y = YPTS(K)
          Z = ZED(K)
          XX = X - XMEAN
          YY = Y - YMEAN
          XR = XX*COS(THETAROT) - YY*SIN(THETAROT) + XMEAN
          YR = YY*COS(THETAROT) + XX*SIN(THETAROT) + YMEAN
          XROT(K) = XR
          YROT(K) = YR
          IF(XR.LT.XMIN) XMIN = XR
          IF(YR.LT.YMIN) YMIN = YR
          IF(XR.GT.XMAX) XMAX = XR
          IF(YR.GT.YMAX) YMAX = YR
          IR = NINT(XR)
          JR = NINT(YR)
          PICTURROT(IR,JR) = Z
          SPICTUR(IR,JR) = Z
          MAP(IR,JR) = Z
          ILOG = (Z/SCALE)
          MAPL(IR,JR) = LOG(1)
    4   CONTINUE
C CALCULATING IMIN, IMAX, JMIN AND JMAX.
        IMIN = NINT(XMIN)
        IMAX = NINT(XMAX)
        JMIN = NINT(YMIN)
        JMAX = NINT(YMAX)
C
        DO 5 K=1,KPTS
          XPTS(K) = XROT(K)
    5     YPTS(K) = YROT(K)
C
        REWIND IUNIT
        DO 6 J=1,KOL
    6     WRITE(IUNIT)(MAP(I,J),I=1,IROW)
        CLOSE(IUNIT)
C
        REWIND JUNIT
        DO 7 J=1,KOL
    7     WRITE(JUNIT)(MAPL(I,J),I=1,IROW)
        CLOSE(JUNIT)
```

```
C
      IW = 0
      IF(IW.EQ.0) GO TO 88
      WRITE(3,112)
      DO 87 I=1,IROW
        WRITE(3,110)I,(PICTURROT(I,J),J=1,20)
        WRITE(3,111)(PICTURROT(I,J),J=21,40)
        WRITE(3,111)(PICTURROT(I,J),J=41,60)
        WRITE(3,111)(PICTURROT(I,J),J=61,80)
        WRITE(3,111)(PICTURROT(I,J),J=81,100)
        WRITE(3,111)(PICTURROT(I,J),J=101,120)
        WRITE(3,111)(PICTURROT(I,J),J=121,128)
   87   CONTINUE
   88   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SMOOTHIMG(NSMOOTH,IROW,KOL,MAXROW,MAXKOL,
     *IMIN,IMAX,JMIN,JMAX,PICTUR,SPICTUR,ZMAX)
C
C     THIS SUBROUTINE READS A REAL MATRIX PICTUR AND OUTPUTS
C A REAL MATRIX 'SPICTUR' WHICH IS THE  SMOOTHED  VERSION OF
C 'PICTUR'.  SMOOTHING IS DONE BY  TAKING  THE  AVERAGE OVER
C A 3X3, 5X5 OR 7X7 WINDOW.   IT ALSO CALCULATES THE LARGEST
C ELEMENT OF THE SMOOTHED MATRIX.
C
C      INPUT DATA TO THE SUBROUTINE.
C IROW    AN INTEGER. IT IS THE NUMBER OF ROWS OF PICTUR.
C KOL     AN INTEGER. IT IS THE NUMBER OF COLUMNS OF PICTUR.
C MAXKOL AN INTEGER .GE. KOL.
C MAXROW AN INTEGER .GE.IROW.
C IMIN, IMAX, JMIN AND JMAX
C         INTEGER  VALUES  WHICH  ARE  THE COORDINATES OF THE
C         RECTANGLE INSIDE WHICH THE OBJECT IS ENCLOSED.
C NSMOOTH AN INTEGER PRAMETER SPECIFIED BY THE USER:   3, 5,
C         OR 7.  IT IS THE SIZE OF THE SMEARING WINDOW.
C PICTUR A REAL 'MAXROW' BY 'MAXKOL' MATRIX WHOSE FIRST IROW
C         ROWS  AND  FIRST  KOL  COLUMNS  CONTAIN  THE  INPUT
C         MATRIX PICTUR.
C
C      OUTPUT OF THE SUBROUTINE.
C SPICTUR A REAL  'MAXROW' BY 'MAXKOL'  MATRIX  WHOSE  FIRST
C         IROW ROWS AND FIRST KOL COLUMNS CONTAIN  THE OUTPUT
C         SMOOTHED MATRIX SPICTUR.
C ZMAX    THE LARGEST SPICTUR(I,J) ELEMENT.
C
      DIMENSION PICTUR(MAXROW,MAXKOL),SPICTUR(MAXROW,MAXKOL)
C
      ZMAX = -1.0
      NS2 = NSMOOTH*NSMOOTH
      ITRIM = (NSMOOTH - 1)/2
      ITRIMP1 = ITRIM + 1
      LIMKOL = KOL - ITRIM
      LIMROW = IROW - ITRIM
      JS = JMIN
      JE = JMAX
      IS = IMIN
      IE = IMAX
      IF(JS.LT.ITRIMP1) JS = ITRIMP1
      IF(IS.LT.ITRIMP1) IS = ITRIMP1
      IF(JE.GT.LIMKOL) JE = LIMKOL
      IF(IE.GT.LIMROW) IE = LIMROW
      DO 4 J=JS,JE
        K1 = J - ITRIM
        K2 = J + ITRIM
        DO 3 I=IS,IE
          L1 = I - ITRIM
          L2 = I + ITRIM
```

```
      SUM = 0.0
      DO 2 K=K1,K2
        DO 1 L=L1,L2
1          SUM = SUM + PICTUR(L,K)
2       CONTINUE
      SUM = SUM/FLOAT(NS2)
      SPICTUR(I,J) = SUM
      IF(SUM.GT.ZMAX) ZMAX = SUM
3     CONTINUE
4   CONTINUE
  RETURN
  END
```

```
      SUBROUTINE DISPLAY(IROW,KOL,MAXROW,MAXKOL,
     *PICTUR,KPTS,XPTS,YPTS,ZMAX,ZED,MATCHARD)
C
C     THIS SUBROUTINE  DISPLAYS  A  GIVEN PICTURE  FILE SUCH
C THAT EACH OF ITS ELEMENTS IS ONE OF THE CHARACTERS '012345
C 6789 ' IN AN ASCENDING ORDER.
C
C      INPUT TO THE SUBROUTINE.
C IROW    AN INTEGER, THE NUMBER OF ROWS OF PICTUR(I,J).
C KOL     AN INTEGER, THE NUMBER OF COLUMNS OF PICTUR(I,J).
C MAXKOL AN INTEGER .GT. KOL.
C MAXROW AN INTEGER .GE.IROW.
C PICTUR A REAL 'MAXROW' BY 'MAXKOL' MATRIX WHOSE FIRST IROW
C        ROWS  AND  FIRST  KOL  COLUMNS  CONTAIN  THE  INPUT
C        MATRIX PICTUR(I,J).
C XPTS    A  REAL  VECTOR  OF LENGTH MAXRK.   ITS  FIRST KPTS
C        ELEMENTS CONTAIN THE X- COORDINATES  OF THE NONZERO
C        POINTS OF THE OBJECT MATRIX.
C YPTS    A  REAL  VECTOR  OF LENGTH MAXRK.   ITS  FIRST KPTS
C        ELEMENTS CONTAIN THE Y- COORDINATES  OF THE NONZERO
C        POINTS OF THE OBJECT MATRIX.
C ZED     A  REAL  VECTOR  OF LENGTH MAXRK.   ITS  FIRST KPTS
C        ELEMENTS CONTAIN THE Z- COORDINATES  OF THE NONZERO
C        POINTS OF THE OBJECT MATRIX.
C ZMAX    A REAL NUMBER, THE LARGEST ELEMENT PICTUR(I,J).
C
C      OUTPUT OF THE SUBROUTINE.
C MATCHARD A CHARACTER*1 MAXROW BY MAXKOL MATRIX WHOSE FIRST
C        IROW ROWS AND FIRST KOL COLUMNS CONTAIN THE  OUTPUT
C        MATRIX, ONE OF THE CHARACTERS '0123456789 '.
C
      DIMENSION PICTUR(MAXROW,MAXKOL)
C
      CHARACTER*1 MATCHARD(MAXROW,MAXKOL)
      CHARACTER STRING*11
      DATA STRING/'0123456789 '/
C
  107 FORMAT(40HO  THE MATRIX CHARACTER MAP             )
  128 FORMAT(1X,128A1)
C
      DO 2 J=1,KOL
        DO 1 I=1,IROW
    1     MATCHARD(I,J) = STRING(11:11)
    2   CONTINUE
        DZ = ZMAX/10.0
        DZ2 = DZ + DZ
        DZ3 = DZ2 + DZ
        DZ4 = DZ2 + DZ2
        DZ5 = DZ3 + DZ2
        DZ6 = DZ3 + DZ3
        DZ7 = DZ4 + DZ3
```

```
      DZ8 = DZ4 + DZ4
      DZ9 = DZ5 + DZ4
      DO 3 K=1,KPTS
        I = XPTS(K)
        J = YPTS(K)
        Z = ZED(K)
        IF(Z.LE.0.0) GO TO 3
        IF(Z.GT.0.0.AND.Z.LE.DZ) MATCHARD(I,J)=STRING(1:1)
        IF(Z.GT.DZ.AND.Z.LE.DZ2) MATCHARD(I,J)=STRING(2:2)
        IF(Z.GT.DZ2.AND.Z.LE.DZ3) MATCHARD(I,J)=STRING(3:3)
        IF(Z.GT.DZ3.AND.Z.LE.DZ4) MATCHARD(I,J)=STRING(4:4)
        IF(Z.GT.DZ4.AND.Z.LE.DZ5) MATCHARD(I,J)=STRING(5:5)
        IF(Z.GT.DZ5.AND.Z.LE.DZ6) MATCHARD(I,J)=STRING(6:6)
        IF(Z.GT.DZ6.AND.Z.LE.DZ7) MATCHARD(I,J)=STRING(7:7)
        IF(Z.GT.DZ7.AND.Z.LE.DZ8) MATCHARD(I,J)=STRING(8:8)
        IF(Z.GT.DZ8.AND.Z.LE.DZ9) MATCHARD(I,J)=STRING(9:9)
        IF(Z.GT.DZ9) MATCHARD(I,J)=STRING(10:10)
    3   CONTINUE
C
      WRITE(3,107)
        DO 21 I=1,IROW
   21   WRITE(3,128)(MATCHARD(I,J),J=1,KOL)
C
      RETURN
      END
```

```
      SUBROUTINE MOMENTAXS(KPTS,MAXRK,XPTS,YPTS,ZED,VOLUM,
     1XMEAN,YMEAN,ZMEAN,ZSIGMA,IMOMNTMAX,IMOMNTMIN,SIGMAMMT,
     2THETAMMT,EPS)
C
C     THIS  SUBROUTINE INPUTS A SET OF REAL NONEZER  Z-COOR-
C DINATES OF THE OBJECT FOR EACH (X,). THEN IT OBTAINS THE X
C AND Y OBJECT COORDINATES 'XMEAN' AND 'YMEAN',  CALCULATES
C THE MAXIMUM AND MINIMUM MOMENTS OF INERTIA 'IMOMNTMAX, AND
C 'IMOMNTMIN'  AT  (XMEAN,YMEAN)  AND  THE  ANGLE 'THETAMMT'
C WHICH THE AXIS OF MAXIMUM MOMENT OF INERTIA MAKES WITH THE
C I-AXIS. ALSO IT CALCULATES THE PARAMETER 'SIGMAMMT'.
C
C      INPUT DATA TO THE SUBROUTINE.
C KPTS   AN INTEGER. IT IS THE NUMBER OF NONZERO ELEMENTS OF
C        THE OBJECT MATRIX.
C MAXRK  AN INTEGER = MAXROW*MAXKOL; 128*128.
C XPTS   A REAL VECTOR  OF  LENGTH  MAXRK.  ITS  FIRST  KPTS
C        ELEMENTS  ARE  THE  X-COORDINATES  OF  THE  NONZERO
C        ELEMENTS OF THE OBJECT MATRIX
C YPTS   A REAL  VECTOR  OF  LENGTH  MAXRK.  ITS FIRST  KPTS
C        ELEMENTS  ARE  THE  Y-COORDINATES  OF  THE  NONZERO
C        ELEMENTS OF THE OBJECT MATRIX.
C ZED    AN REAL  VECTOR  OF  LENGTH MAXRK.  ITS FIRST  KPTS
C        ELEMENTS ARE THE NONZERO ELEMENTS OF OBJECT MATRIX.
C EPS    A SPECIFIED TOLERANCE SUCH THAT A CALCULATED NUMBER
C        Z IS CONSIDERED ZERO IF |Z|<EPS. FOR THE VAX 11/780
C        COMPUTER, EPS IS USUALLY TAKEN 1.E-4 AND 1.E-11 FOR
C        SINGLE AND DOUBLE PRECISION RESPECTIVELY.
C
C      OUTPUT OF THE SUBROUTINE.
C VOLUM  A REAL NUMBER WHICH IS THE VOLUME OF THE OBJECT.
C XMEAN  A REAL NUMBER WHICH IS THE MEAN VALUE OF THE X  CO-
C        ORDINATES OF THE OBJECT.
C YMEAN  A REAL NUMBER WHICH IS THE MEAN VALUE OF THE Y  CO-
C        ORDINATE OF THE OBJECT.
C ZMEAN  A REAL NUMBER WHICH IS THE MEAN VALUE OF THE Z  CO-
C        ORDINATES OF THE OBJECT.
C ZSIGMA A REAL  NUMBER  WHICH  IS THE STANDARD DEVIATION OR
C        THE SQUARE ROOT OF THE VARIANCE OF THE Z CO-ORDINA-
C        TES OF THE OBJECT.
C IMOMNTMAX  AN INTEGER,  THE  MAXIMUM  MOMENT OF INERTIA OF
C        THE OBJECT ABOUT AN AXIS THROUGH (XMEAN,YMEAN).
C IMOMNTMIN  AN INTEGER,  THE  MINIMUM  MOMENT OF INERTIA OF
C        THE OBJECT ABOUT AN AXIS THROUGH (XMEAN,YMEAN).
C THETAMMT   A REAL NUMBER.  IT  IS  THE  ANGEL (IN RADIANS)
C        BETWEEN  THE  AXIS OF MAXIMUM MOMENT OF INERTIA AND
C        THE PRESENT X AXIS.
C SIGMAMMT  A REAL INVARIANT PARAMETER.
C
      DIMENSION XPTS(MAXRK),YPTS(MAXRK),ZED(MAXRK)
C
```

```
      PID4 = ATAN(1.0)
      VOLUM = 0.0
      XMEAN = 0
      YMEAN = 0
      DO 1 K=1,KPTS
        Z = ZED(K)
        VOLUM = VOLUM + Z
        XMEAN = XMEAN + Z*XPTS(K)
    1   YMEAN = YMEAN + Z*YPTS(K)
C VOLUM, XMEAN AND YMEAN.
      XMEAN = XMEAN/VOLUM
      YMEAN = YMEAN/VOLUM
      ZMEAN = VOLUM/FLOAT(KPTS)
C
      EXY = 0.0
      EXX = 0.0
      EYY = 0.0
      ZVAR = 0.0
      DO 2 K=1,KPTS
        Z = ZED(K)
        DX = (XPTS(K) - XMEAN)
        DY = (YPTS(K) - YMEAN)
        DZ = (ZED(K) - ZMEAN)
        EXY = EXY + DX*DY*Z
        EXX = EXX + DY*DY*Z
        EYY = EYY + DX*DX*Z
    2   ZVAR = ZVAR + DZ*DZ
C
      SIGMAX = SQRT(EXX/VOLUM)
      SIGMAY = SQRT(EYY/VOLUM)
      SIGMAMMT = SQRT((EXX + EYY)/VOLUM)
      ZVAR = ZVAR/FLOAT(KPTS)
      ZSIGMA = SQRT(ZVAR)
      SUM1 = 0.0
      SUM2 = EXY + EXY
      SUM3 = EXX + EYY
      DEL = ABS(EXX - EYY)
      IF(DEL.GT.EPS) GO TO 3
      ANGELI = PID4
      GO TO 4
    3 SUM1 = EXX - EYY
      THETAMMT = 0.5*ATAN(SUM2/SUM1)
    4 SUMMAX = 0.5*(SUM3 + SQRT(SUM1*SUM1 + SUM2*SUM2))
      SUMMIN = 0.5*(SUM3 - SQRT(SUM1*SUM1 + SUM2*SUM2))
      IMOMNTMAX = SUMMAX
      IMOMNTMIN = SUMMIN
      RETURN
      END
```

```
      SUBROUTINE EQUATION(MAXRK,MAXX,KOLX,KPTS,XMEAN,YMEAN,
     *C,XPTS,YPTS)
C
C     THIS SUBROUTINE CALCULATES THE COEFFICIENT MATRIX   OF
C AN OVERDETERMINED SYSTEM OF LINEAR EQUATIONS.    THE LEAST
C SQUARES SOLUTION OF THIS EQUATION GIVES  THE  COEFFICIENTS
C OF THE EQUATION OF THE  ELLIPTIC  (HYPERBOLIC)  PARABOLOID
C WHICH FITS THE NONZERO  Z-COORDINATES OF THE OBJECT IN THE
C LEAST SQUARES SENSE.
C     THE EQUATION OF THE ELLIPTIC  (HYPERBOLIC)  PARABOLOID
C SURFACE IS
C      Z=A(1)+A(2)*X+A(3)*Y+A(4)*X*X+A(5)*X*Y+A(6)*Y*Y    (A)
C OR
C              Z=A(1)+A(4)*X*X+A(5)*X*Y+A(6)*Y*Y          (B)
C
C      INPUT DATA TO THE SUBROUTINE.
C MAXRK  AN INTEGER = MAXROW*MAXKOL = 128*128.
C MAXX   AN INTEGER .GT.  THE  MAXIMUM  NUMBER OF COLUMNS IN
C        THE COEFFICIENT MATRIX. WE TAKE MAXX=7.
C KPTS   AN INTEGER. LE. MAXRK.  IT IS THE NUMBER OF ROWS IN
C        IN THE EQUATION TO BE SOLVED.
C KOLX   AN INTEGER .EQ.NUMBER OF COLUMNS IN THE COEFFICIENT
C        MATRIX.  FOR  EQUATION (A) ABOVE,  KOLX=6  AND  FOR
C        EQUATION (B) KOLX=4.
C XPTS   A REAL VECTOR OF LENGTH MAXRK.    ITS FIRST  'KPTS'
C        CONTAIN THE X-COORDINATES TO BE SUBSTITUTED IN  THE
C        EQUATION OF THE SURFACE (A) OR (B).
C XPTS   A REAL VECTOR OF LENGTH MAXRK.    ITS FIRST  'KPTS'
C        CONTAIN THE Y-COORDINATES TO BE SUBSTITUTED IN  THE
C        EQUATION OF THE SURFACE (A) OR (B).
C XMEAN  A REAL NUMBER WHICH  IS  THE  X   COORDINATE OF THE
C        CENTER OF GRAVITY OF THE OBJECT.
C YMEAN  A REAL NUMBER WHICH  IS  THE  Y   COORDINATE OF THE
C        CENTER OF GRAVITY OF THE OBJECT.
C
C      OUTPUT OF THE SUBROUTINE.
C C      A REAL MAXRK BY MAXX MATRIX. ITS FIRST  'KPTS' ROWS
C        AND FIRST 'KOLX' COLUMNS  CONTAIN  THE  COEFFICIENT
C        MATRIX OF THE APPROXIMATING EQUATION.
C
      DIMENSION XPTS(MAXRK),YPTS(MAXRK),C(MAXRK,MAXX)
C
      IF(KOLX.EQ.6) GO TO 2
      DO 1 K=1,KPTS
        AI = XPTS(K) - XMEAN
        AJ = YPTS(K) - YMEAN
        C(K,1) = 1.0
        C(K,2) = AI*AI
        C(K,3) = AI*AJ
    1   C(K,4) = AJ*AJ
      GO TO 4
```

```
2 DO 3 K=1,KPTS
    AI = XPTS(K) - XMEAN
    AJ = YPTS(K) - YMEAN
    C(K,1) = 1.0
    C(K,2) = AI
    C(K,3) = AJ
    C(K,4) = AI*AI
    C(K,5) = AI*AJ
3   C(K,6) = AJ*AJ
4 CONTINUE
  RETURN
  END
```

```
      SUBROUTINE NORMEQLS(KPTS,KOLX,MAXRK,MAXX,A,B,X,R,
     *EL,Y,EPS,RES,RHO,IND)
C
C     GIVEN IS THE REAL  'KPTS' BY 'KOLX' MATRIX 'A' OF RANK
C KOLX.LE.KPTS AND A REAL 'KPTS' VECTOR 'B' FOR THE EQUATION
C                          A*X = B.
C     THIS SUBROUTINE OBTAINS THE NORMAL EQUATION FOR  A*X=B
C THEN CALCULATES THE LEAST SQUARES SOLUTION X OF THE  OVER-
C DETERMINED SYSTEM A*X=B.  FOR SOLVING THE NORMAL EQUATION,
C MATRIX  [A(TRANSPOSE)*A]  IS  FACTORIZED  BY  THE CHOLESKY
C DECOMPOSITION METHOD.
C
C      INPUT DATA TO SUBROUTINE.
C KPTS    AN INTEGER, THE NUMBER OF ROWS OF MATRIX A.
C KOLX    AN INTEGER, THE NUMBER OF COLUMNS OF MATRIX A.
C MAXX    AN INTEGER .GE. (KOLX+1).
C MAXRK   AN INTEGER .GE.KPTS, MAXRK.GE.MAXX.
C A       A REAL 'MAXRK BY MAXX' MATRIX. ITS FIRST KPTS  ROWS
C         AND FIRST KOLX COLUMNS  CONTAIN  MATRIX 'A' OF  THE
C         EQUATION A*X=B. MATRIX 'A' IS NOT DESTROYED  IN THE
C         COMPUTATION.
C B       A REAL MAXRK VECTOR WHOSE FIRST KPTS CONTAIN VECTOR
C         'B' OF THE GIVEN EQUATION A*X=B.    VECTOR B IS NOT
C         DESTROYED BY THE COMPUTATION.
C EPS     A SPECIFIED TOLERANCE SUCH THAT A CALCULATED NUMBER
C         Z IS CONSIDERED ZERO IF |Z|<EPS. FOR THE VAX 11/780
C         COMPUTER, EPS IS USUALLY TAKEN 1.E-4 AND 1.E-11 FOR
C         SINGLE AND DOUBLE PRECISION RESPECTIVELY.
C
C     OUTPUT OF THE SUBROUTINE.
C X       A REAL 'MAXX' VECTOR  WHOSE FIRST KOLX ELEMENTS ARE
C         THE  LEAST SQUARES SOLUTION OF THE SYSTEM A*X=B.
C RES     A REAL  MAXRK  VECTOR WHOSE FIRST KPTS ELEMENTS ARE
C         THE  RESIDUAL VECTOR (B-A*X) OF A*X=B.
C RHO     A REAL NUMBER WHICH IS THE L2 NORM OF THE  RESIDUAL
C         VECTOR RES.
C IND     AN INTEGER INDICATOR. IND=0 INDICATES  NORMAL EXIT.
C         IND=-1 INDICATES THAT KPTS.LT.KOLX, THIS SUBROUTINE
C         IS NOT DESIGNED FOR UNDERDETERMINED SYSTEMS,    OR,
C         MAXX.LT.(KOLX+1), OR MAXX.GT.MAXRK.   IND=-2 IS FOR
C         PREMATURE EXIT DUE TO RANK DEFICIENCY OF MATRIX A.
C
C     THE MEANING OF SOME OTHER DIMENSIONED PARAMETERS.
C R       A REAL 'MAXX BY MAXX'   MATRIX WHOSE FIRST (KOLX+1)
C         ROWS AND FIRST 'KOLX' COLUMNS CONTAIN  THE  PRODUCT
C         A(TRANSPOSE)*(A|B).
C Y       A REAL MAXX WORKING VECTOR.
C EL      A REAL 'MAXX BY MAXX'  MATRIX.  ITS FIRST KOLX ROWS
C         AND FIRST KOLX COLUMNS CONTAIN THE LOWER TRIANGULAR
C         MATRIX  IN THE  CHOLESKY DECOMPOSITION OF THE FIRST
C         KOLX COLUMNS OF MATRIX R WHICH IS  EL*EL(TRANSPOSE)
```

```
C
      DOUBLE PRECISION S,SA
      DIMENSION A(MAXRK,MAXX),B(MAXRK),X(MAXX),R(MAXX,MAXX),
     1EL(MAXX,MAXX),Y(MAXX),RES(MAXRK)
C
  120 FORMAT(40H0 ** EXIT SUBROUTINE NORMEQLS****,IND = ,I8)
C INITIALIZATION.
      IND = 0
      IF(KOLX.GT.KPTS) GO TO 22
      IF(MAXX.LE.KOLX) GO TO 22
      IF(MAXRK.LT.MAXX) GO TO 22
      KOLXP1 = KOLX + 1
      DO 3 J=1,KOLX
        DO 2 I=J,KOLX
          S = 0.0
          DO 1 K=1,KPTS
    1       S = S + A(K,I)*A(K,J)
          R(J,I) = S
    2     R(I,J) = S
    3   CONTINUE
      DO 5 I=1,KOLX
        S = 0.0
        DO 4 K=1,KPTS
    4     S = S + A(K,I)*B(K)
    5   R(I,KOLXP1) = S
C CHOLESKY DECOMPOSITION OF POSITIVE DEFINITE
C   SYMMETRIC MATRIX.
      IRNK = KOLX
      L = 1
      C = R(1,1)
      IF(C.LT.EPS) GO TO 23
      EL(1,1) = C**0.5
      IF (IRNK.LE.1) GO TO 12
    6 LP1 = L+1
      EL(LP1,1) = R(1,LP1)/EL(1,1)
      IF (L.EQ.1) GO TO 9
      DO 8 I=2,L
        IM1 = I - 1
        CONST = -R(I,LP1)
        SM = CONST
        DO 7 K=1,IM1
          SMA = EL(I,K)
          SMB = EL(LP1,K)
    7     SM = SM + SMA*SMB
        GH = SM
    8   EL(LP1,I) = -GH/EL(I,I)
    9 DO 10 I=1,L
   10   EL(I,LP1) = 0.0
      CONST = - R(LP1,LP1)
      SM = CONST
      DO 11 K=1,L
```

```
          SMA = EL(LP1,K)
   11     SM = SM + SMA*SMA
          RGH = SM
          EL(LP1,LP1) = (-RGH)**0.5
          L = L + 1
          IF (L.LT.IRNK) GO TO 6
C CALCULATING VECTOR Y: Y = EL*X.
   12 Y(1) = R(1,KOLXP1)/EL(1,1)
          IF(KOLX.EQ.1) GO TO 15
          DO 14 II=2,KOLX
               IIM1 = II - 1
               S = - R(II,KOLXP1)
               DO 13 K=1,IIM1
   13            S = S + EL(II,K)*Y(K)
   14       Y(II) = - S/EL(II,II)
C CALCULATING VECTOR X.
   15 X(KOLX) = Y(KOLX)/EL(KOLX,KOLX)
          IF(KOLX.EQ.1) GO TO 18
          DO 17 II=2,KOLX
               I = KOLX - II + 1
               IP1 = I + 1
               S = -Y(I)
               DO 16 K=IP1,KOLX
   16            S = S + EL(K,I)*X(K)
   17          X(I) = -S/EL(I,I)
   18 CONTINUE
C CALCULATING THE RESIDUAL VECTOR.
          RESMAX = -1.0
          SA = 0.0
          DO 21 J=1,KPTS
            S = -B(J)
            DO 19 K=1,KOLX
   19         S = S + A(J,K)*X(K)
            SS = S
            IF(SS.LT.0.0) SS = -SS
            IF(SS.LT.RESMAX) GO TO 20
            RESMAX = SS
            JRESMAX = J
   20       RES(J) = -S
            SA = SA + S*S
   21       CONTINUE
          RHO = SA**0.5
          GO TO 24
   22 IND = -1
          WRITE(3,120)IND
          GO TO 24
   23 IND = -2
          WRITE(3,120)IND
   24 CONTINUE
          RETURN
          END
```

```
      SUBROUTINE ELLIPSE(KOLX,MAXX,A,ANEW,ASEMIMAJ,
     *BSEMIMIN,THETAELL,EPS,ECCENT,INDE)
C
C     GIVEN IS A GENERAL EQUATION OF AN ELLIPSE (HYPERBOLA)
C THIS SUBROUTINE CALCULATES THE SEMI-MAJOR AND  SEMI-MINOR
C AXES OF THE CONIC AND THE ANGLE THE MAJOR AXIS MAKES WITH
C THE X-AXIS. THE GIVEN EQUATION OF THE ELLIPSE (HYPERBOLA)
C IS
C     0=A(1)+A(2)*X+A(3)*Y+A(4)*X*X+A(5)*X*Y+A(6)*Y*Y
C OR
C          0=A(1)+A(4)*X*X+A(5)*X*Y+A(6)*Y*Y.
C WE WISH TO TRANSFORM THIS EQUATION TO THE STANDARD FORM
C           0=ANEW(1)+ANEW(4)*X*X+ANEW(6)*Y*Y
C FROM WHICH THE MAJOR AND MINOR SEMI-AXES AND THE ECCENTRI-
C CITY OF THE CONIC ARE OBTAINED.
C
C      INPUT TO THE SUBROUTINE.
C KOLX   AN INTEGER =  NUMBER OF COEFFICIENTS OF THE ELLIPSE
C        (HYPERBOLA), (IN OUR CASE IT IS EITHER 6 OR 4).
C MAXX   AN INTEGER .GT. KOLX. WE HERE TAKE MAXX=7.
C A      A REAL VECTOR OF LENGTH MAXX. ITS FIRST KOLX
C        ELEMENTS ARE THE COEFFICIENTS OF THE EQUATION OF
C        THE ELLIPSE (HYPERBOLA).
C EPS    A SPECIFIED TOLERANCE SUCH THAT A CALCULATED NUMBER
C        Z IS CONSIDERED ZERO IF |Z|<EPS. FOR THE VAX 11/780
C        COMPUTER, EPS IS USUALLY TAKEN 1.E-4 AND 1.E-11 FOR
C        SINGLE AND DOUBLE PRECISION RESPECTIVELY.
C
C      OUTPUT OF THE SUBROUTINE.
C ANEW   A REAL VECTOR OF LENGTH MAXX. ITS FIRST KOLX
C        ELEMENTS ARE THE COEFFICIENTS OF THE EQUATION OF
C        A STANDARD ELLIPSE (HYPERBOLA).
C ASEMIMAJ A REAL NUMBER, THE SEMI MAJOR AXIS OF THE CONIC.
C BSEMIMIN A REAL NUMBER, THE SEMI MINOR AXIS OF THE CONIC.
C THETAELL A REAL NUMBER.  IT IS THE  ANGLE (IN RADIANS) THE
C        MAJOR  AXIS  OF  THE ELLIPSE (HYPERBOLA) MAKES WITH
C        THE X-AXIS.
C ECCENT THE  CALCULATED  ECCENTRICITY OF THE ELLIPSE OR THE
C        HYPERBOLA.
C INDE   AN INTEGER INDICATOR.    INDE=0 INDICATES  THAT THE
C        GIVEN EQUATION IS THAT OF AN ELLIPSE. INDE=1  INDI-
C        CATES THAT THE GIVEN  EQUATION IS  OF A  HYPERBOLA,
C        INDE=-1 INDICATES THAT THE G IVEN  EQUATION  IS FOR
C        NEITHER AN ELLIPSE NOR  FOR  A  HYPERBOLA  AND  THE
C        SUBROUTINE EXITS PREMATURELY.
C
      DIMENSION A(MAXX),ANEW(MAXX)
C
      INDE = 0
      PI = 4.0*ATAN(1.0)
      THETAELL = PI/4.0
```

```
      IF(KOLX.EQ.4) GO TO 3
      CONX = A(3)*A(5) - 2.0*A(2)*A(6)
      CONY = 2.0*A(3)*A(4) - A(2)*A(5)
      CONO = 4.0*A(4)*A(6) - A(5)*A(5)
      ACONX = ABS(CONX)
      IF(ACONX.LT.EPS) GO TO 1
      CON = CONO/CONX
      IF(CON.LT.0.0) CON = - CON
      IF(CON.LT.EPS) GO TO 11
    1 ACONY = ABS(CONY)
      IF(ACONY.LT.EPS) GO TO 2
      CON = CONO/CONY
      IF(CON.LT.0.0) CON = - CON
      IF(CON.LT.EPS) GO TO 11
C XCE, YCE  TWO REAL NUMBER,  THE  X-  AND  Y- COORDINATE
C      OF THE CENTER OF THE ELLIPSE (HYPERBOLA).
    2 XCE = CONX/CONO
      YCE = - CONY/CONO
      ANEW(1) = A(1) + XCE*A(2) + YCE*A(3) + XCE*XCE*A(4)
     *                          + XCE*YCE*A(5) + YCE*YCE*A(6)
      GO TO 4
    3 A(6) = A(4)
      A(5) = A(3)
      A(4) = A(2)
      ANEW(1) = A(1)
    4 ANEW(2) = 0.0
      ANEW(3) = 0.0
      ANEW(4) = A(4)
      ANEW(6) = A(6)
      IF(A(5).EQ.0.0) GO TO 6
      IF(A(5).EQ.0.0) THETAELL = 0.0
      CONA = A(4) - A(6)
      SUM1 = A(5)
      SUM2 = A(4) - A(6)
      SUM3 = A(4) + A(6)
      IF(CONA.LT.0.0) CONA = - CONA
      IF(CONA.LT.EPS) GO TO 5
      RATIO = SUM1/SUM2
      THETAELL = 0.5*ATAN(RATIO)
    5 ANEW(4) = 0.5*(SUM3 + SQRT(SUM1*SUM1 + SUM2*SUM2))
      ANEW(6) = 0.5*(SUM3 - SQRT(SUM1*SUM1 + SUM2*SUM2))
    6 ANEW(5) = 0.0
      A1 = - ANEW(1)/ANEW(4)
      B1 = - ANEW(1)/ANEW(6)
      C1 = A1*B1
      IF(C1.LT.0.0) GO TO 8
      IF(A1.GT.B1) GO TO 7
      CONAB = 1.0 - (A1/B1)
      ASEMIMAJ = SQRT(B1)
      BSEMIMIN = SQRT(A1)
      ECCENT = SQRT(CONAB)
```

```
      GO TO 12
    7 CONAB = 1.0 - (B1/A1)
      ASEMIMAJ = SQRT(A1)
      BSEMIMIN = SQRT(B1)
      ECCENT = SQRT(CONAB)
      GO TO 12
    8 INDE = 1
      IF(A1.GT.0.0) GO TO 9
      CONAB = 1.0 + (-A1/B1)
      ASEMIMAJ = SQRT(B1)
      BSEMIMIN = SQRT(-A1)
      ECCENT = SQRT(CONAB)
      GO TO 12
    9 CONAB = 1.0 + (-B1/A1)
      ASEMIMAJ = SQRT(A1)
      BSEMIMIN = SQRT(-B1)
      ECCENT = SQRT(CONAB)
      GO TO 12
   11 INDE = -1
   12 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE KONTOURS(IMIN,IMAX,JMIN,JMAX,IROW,KOL,
     1MAXROW,MAXKOL,MAXRAK,SPICTUR,KONTXPTS,KONTYPTS,
     2KONTPTS,MATCHAR,IKONT,EPS)
C
C     THIS SUBROUTINE  (1) FINDS OUT THE OUTSIDE  AND INSIDE
C CONTOURS OF A TWO DIMENSIONAL IMAGE USING A SIMPLE METHOD.
C (2) IT WRITES  DOWN THE I AND J-COORDINATES OF THE CONTOUR
C OF THE OBJECT IN THE FORM OF A STRING OF THE I-COORDINATES
C AND THE CORRESPONDING STRING  OF  THE  J-COORDINATES,  AND
C (3) THE  NUMBER OF THE OBTAINED CONTOUR POINTS.
C
C     THE METHOD OF FINDING A CONTOUR POINT IS AS FOLLOWS.
C THE IMAGE MATRIX IS SCANNED COLUMN WISE. CONSIDER NON-ZERO
C ELEMENTS.   EACH  OF  THE  8 NEIGHBOURING ELEMENTS TO THIS
C NON-ZERO ELEMENT AS A CENTER IS EXAMINED. IF ANY OF THE  8
C ELEMENTS IS ZERO, THEN THE CENTER ELEMENT IS A BOUNDARY OR
C A CONTOUR POINT.
C
C     INPUT DATA TO THE SUBROUTINE.
C IMIN, IMAX, JMIN AND JMAX : INTEGER PARAMETERS, THE  COOR-
C        MATES OF  THE  RECTANGLE INSIDE WHICH THE OBJECT IS
C        ENCLOSED.
C IROW   AN INTEGER. IT IS THE NUMBER OF ROWS OF SPICTUR.
C KOL    AN INTEGER. IT IS THE NUMBER OF COLUMNS OF SPICTUR.
C MAXKOL AN INTEGER .GE. KOL.
C MAXROW AN INTEGER .GE.IROW.
C MAXRAK AN INTEGER = 4*(MAXROW + MAXKOL)
C SPICTUR AN MAXROW BY MAXKOL OBJECT MATRIX.
C IKONT  AN INTEGER SPECIFIED BY THE USER.   FOR IKONT.EQ.0,
C        THE SUBROUTINE DOES NOT DISPLAY THE OBJECT CONTOUR,
C        FOR IKONT.NE.0, THE OBJECT CONTOUR IS DISPLAYED  IN
C        BINARY FORM, '1'S AND ' 'S.
C EPS    A SPECIFIED TOLERANCE SUCH THAT A CALCULATED NUMBER
C        X IS CONSIDERED ZERO IF |X|<EPS. FOR THE VAX 11/780
C        COMPUTER, EPS IS USUALLY TAKEN 1.E-4 AND 1.E-11 FOR
C        SINGLE AND DOUBLE PRECISION RESPECTIVELY.
C
C      OUTPUT OF THE SUBROUTINE.
C KONTPTS AN INTEGER.  IT IS THE NUMBER OF KONTOUR POINTS OF
C        THE GIVEN IMAGE.
C KONTXPTS AN INTEGER VECTOR  OF  LENGTH  MAXRK.  ITS  FIRST
C        KONTPTS ELEMENTS ARE THE  (X-) I-COORDINATES OF THE
C        KONTOUR POINTS.
C KONTYPTS AN INTEGER VECTOR  OF  LENGTH  MAXRK.  ITS  FIRST
C        KONTPTS ELEMENTS ARE THE  (Y-) J-COORDINATES OF THE
C        KONTOUR POINTS WHICH CORRESPOND TO 'KONTXPTS'.
C MATCHAR  A CHARACTER*1 MAXROW BY MAXKOL MATRIX WHOSE FIRST
C        IROW ROWS AND FIRST KOL COLUMNS CONTAIN THE  OUTPUT
C        CHARACTER MATRIX.  EACH OF ITS ELEMENTS IS EITHER A
C        ' ' OR A '1'.
C
```

```
      DIMENSION SPICTUR(MAXROW,MAXKOL),KONTXPTS(MAXRAK),
     *KONTYPTS(MAXRAK)
      CHARACTER*1 MATCHAR(MAXROW,MAXKOL)
C
      CHARACTER STRING*11
      DATA STRING/'0123456789 '/
  128 FORMAT(1X,128A1)
C
      I1 = IMIN
      I2 = IMAX
      J1 = JMIN
      J2 = JMAX
      IF(I1.EQ.1) I1 =2
      IF(J1.EQ.1) J1 =2
      IF(I2.EQ.IROW) I2 = IROW - 1
      IF(J2.EQ.KOL) J2 = KOL - 1
C FILLING GAP POINTS.
      DO 3 J=J1,J2
        DO 2 I=I1,I2
          A0 = SPICTUR(I,J)
          A1 = SPICTUR(I-1,J)
          A2 = SPICTUR(I+1,J)
          IF(A0.GT.EPS) GO TO 2
          IF(A1.GT.EPS.AND.A2.GT.EPS) GO TO 1
          GO TO 2
    1     SPICTUR(I,J) = (A1+A2)/2
    2     CONTINUE
    3   CONTINUE
C FORMING THE CONTOUR MAP OF ' 'S AND '1'S.
      KONTPTS = 0
      DO 6 J=J1,J2
        DO 5 I=I1,I2
          MATCHAR(I,J) = STRING(11:11)
          IF(SPICTUR(I,J).LT.EPS) GO TO 5
          IF(SPICTUR(I-1,J-1).LT.EPS) GO TO 4
          IF(SPICTUR(I-1,J).LT.EPS) GO TO 4
          IF(SPICTUR(I-1,J+1).LT.EPS) GO TO 4
          IF(SPICTUR(I,J+1).LT.EPS) GO TO 4
          IF(SPICTUR(I+1,J+1).LT.EPS) GO TO 4
          IF(SPICTUR(I+1,J).LT.EPS) GO TO 4
          IF(SPICTUR(I+1,J-1).LT.EPS) GO TO 4
          IF(SPICTUR(I,J-1).LT.EPS) GO TO 4
          GO TO 5
    4     MATCHAR(I,J) = STRING(2:2)
          KONTPTS = KONTPTS + 1
          KONTXPTS(KONTPTS) = I
          KONTYPTS(KONTPTS) = J
    5     CONTINUE
    6   CONTINUE
        IF(IKONT.EQ.0) GO TO 8
        DO 7 I=IMIN,IMAX
```

```
      7    WRITE(3,128)(MATCHAR(I,J),J=JMIN,JMAX)
      8 CONTINUE
        RETURN
        END
```

```
      SUBROUTINE MMTAXSKNT(KONTPTS,MAXRAK,KONTXPTS,KONTYPTS,
     *MMTMAXKNT,MMTMINKNT,THETAKNT,SIGMAKNT)
C
C     THIS SUBROUTINE INPUTS A SET OF INTEGER CONTOUR POINTS
C WHOSE NUMBER IS 'KONTPTS' AND WHOSE I AND J COORDINATES
C ARE THE INTEGER ARRAYS KONTXPTS AND KONTYPTS. THE CONTOUR
C FORMS A BINARY IMAGE. THE CONTOUR POINTS HAVE THE VALUE 1
C AND THE REST OF THE IMAGE ZERO.
C     THIS SUBROUTINE THEN OBTAINS THE I AND J COORDINATES
C 'IXCENTKNT AND IYCENTKNT' OF THE CENTER OF MASS OF THE
C CONTOUR POINTS AND CALCULATES THE MAXIMUM AND MINIMUM
C MOMENTS OF INERTIA MMTMAXKNT AND MMTMINKNT AT (IXCENTKNY,
C IYCENTKNT) AND THE ANGLE 'THETAKNT' WHICH THE AXIS OF
C MAXIMUM MOMENT OF INERTIA OF THE CONTOUR POINTS MAKES WITH
C THE I-AXIS.     THEN IT CALCULATES THE INVARIANT PARAMETER
C 'SIGMAKNT'.
C
C      INPUT DATA TO THE SUBROUTINE.
C MAXRAK AN INTEGER = 4*(MAXROW+MAXKOL) = 4*(128+128).
C KONTPTS AN INTEGER. THE NUMBER OF POINTS ON THE CONTOUR.
C KONTXPTS AN INTEGER ARRAY OF LENGTH MAXRAK. ITS FIRST
C        KONTPTS ELEMENTS ARE THE I COORDINATES OF THE
C        CONTOUR POINTS
C KONTYPTS AN INTEGER ARRAY OF LENGTH MAXRAK. ITS FIRST
C        KONTPTS ELEMENTS ARE THE J COORDINATES OF THE
C        CONTOUR POINTS
C
C      OUTPUT OF THE SUBROUTINE.
C IXCENKNT  AN INTEGER WHICH IS THE I-COORDINATE OF THE
C        CENTER OF GRAVITY OF THE CONTOUR.
C JYCENKNT  AN INTEGER WHICH IS THE J-COORDINATE OF THE
C        CENTER OF GRAVITY OF THE CONTOUR.
C MMTMAXKNT AN INTEGER. IT IS THE MAXIMUM MOMENT OF INERTIA
C        OF THE KONTOUR POINTS OF THE OBJECT THROUGH AN AXIS
C        PASSING THROUGH THE CENTER OF MASS.
C MMTMINKNT AN INTEGER. IT IS THE MINIMUM MOMENT OF INERTIA
C        OF THE KONTOUR POINTS OF THE OBJECT THROUGH AN AXIS
C        PASSING THROUGH THE CENTER OF MASS.
C THETAKNT  A REAL NUMBER. IT IS THE ANGEL (IN RADIANS)
C        BETWEEN THE AXIS OF MAXIMUM MOMENT OF INERTIA AND
C        THE PRESENT I AXIS.
C SIGMAKNT A REAL INVARIANT PARAMETER.
C
      DIMENSION KONTXPTS(MAXRAK),KONTYPTS(MAXRAK)
C
C INITIALIZATION.
      PID2 = 2.0*ATAN(1.0)
      IXCENKNT = 0
      JYCENKNT = 0
      DO 1 I=1,KONTPTS
        IXCENKNT = IXCENKNT + KONTXPTS(I)
```

```
1     JYCENKNT = JYCENKNT + KONTYPTS(I)
      IXCENKNT = IXCENKNT/KONTPTS
      JYCENKNT = JYCENKNT/KONTPTS
      IXX = 0
      IYY = 0
      IXY = 0
      DO 2 I=1,KONTPTS
        DX = KONTXPTS(I) - IXCENKNT
        DY = KONTYPTS(I) - JYCENKNT
        IXX = IXX + DX*DX
        IYY = IYY + DY*DY
2     IXY = IXY + DX*DY
C
      CX = IXX/KONTPTS
      CY = IYY/KONTPTS
      SIGMAX = SQRT(CX)
      SIGMAY = SQRT(CY)
      SIGMAKNT = SQRT(CX + CY)
      SUM2 = FLOAT(IXY+IXY)
      SUM3 = FLOAT(IXX+IYY)
      IF(IXX.NE.IYY) GO TO 3
      SUM1 = 0.0
      THETAKNT = PID2
      GO TO 4
3     SUM1 = FLOAT(IYY-IXX)
      THETAKNT = 0.5*ATAN(SUM2/SUM1)
4     SUMMAX = 0.5*(SUM3 + SQRT(SUM1*SUM1 + SUM2*SUM2))
      SUMMIN = 0.5*(SUM3 - SQRT(SUM1*SUM1 + SUM2*SUM2))
      MMTMAXKNT = NINT(SUMMAX)
      MMTMINKNT = NINT(SUMMIN)
      RETURN
      END
```

```
      SUBROUTINE CONVOL5(IROW,KOL,MAXROW,MAXKOL,MAXRAK,
     1NSIDE,SPICTUR,IMIN,IMAX,JMIN,JMAX,XSCALE,YSCALE,
     2FU,FV,FUV,FUU,FVV,EKEE,EHEE,SKEE,SHEE,EDGMAP,
     3FLATMAP,ERRSURF,V,MAPL,IMAPL,ABIGK,ABIGH,IERRORC)
C
C     GIVEN IS A 3-D OBJECT IN THE FORM OF THE  IMAGE MATRIX
C 'SPICTUR' WHOSE (I,J) VALUES ARE THE  Z-COORDINATES OF THE
C OBJECT.
C     THIS SUBROUTINE CALCULATES (1) THE 'SMOOTHED GAUSSIAN'
C 'SKEE(I,J)' AND  (2) THE 'SMOOTHED MEAN' 'SHEE(I,J)'  CUR-
C VATURES  AT EACH  POINT (I,J) OF THE SURFACE SPICTUR(I,J).
C THE SMOOTHED VALUE IS THE AVERAGE OVER A 7 BY 7 WINDOW.
C     IT  ALSO  CALCULATES THE 3 SURFACES:  (3) EDGMAP,  (4)
C FLATMAP AND (5) ERRSURF WHICH ARE RESPECTIVELY INDICATIONS
C OF THE EDGES, THE FLATNESS AND A  MEASURE OF THE ERROR  IN
C THE LEAST SQUARES FIT AT (I,J).
C
C     IF DESIRED BY THE USER, THIS SUBROUTINE MAPS  SURFACES
C (1) SKEE, (2) SHEE, (3) EDGMAP,(4) FLATMAP AND (5) ERRSURF
C ONTO 'LOGICAL*1' FILES UNDER CERTAIN UNIT NUMBERS (94, 95,
C ..., 98 RESPECTIVELY).
C     AT EACH POINT (I,J) OF 'SPICTUR', A  QUADRATIC SURFACE
C OF THE  FORM
C          F(I,J) = A00 + A10*I + A01*J + A11*I*J
C                              + A20*(I*I-2) + A02*(J*J-2).
C IS FIT IN A LEAST SQUARES .
C     ONE CALCULATES THE PARTIAL DERIVATIVES OF SPICTUR(I,J)
C W.R.T. U (I.E OF I) AND V (I,E. OF J).  THESE  ARE DENOTED
C BY FU, FV, FUU, FUV  AND  FVV.  THESE IN FACT ARE OBTAINED
C BY CONVOLVING SPICTUR(U,V) WITH  CERTAIN WEIGHTED WINDOWS.
C SEE: P. R. BEAUDET,  ROTATIONAL INVARIANT IMAGE OPERATORS,
C 4TH  ICPR, NOVEMBER 7-10, 1978, KYOTO, JAPAN, PP. 579-583.
C
C      INPUT DATA TO THE SUBROUTINE.
C IROW    AN INTEGER, NUMBER  OF  ROWS  OF  THE  GIVEN OBJECT
C         MATRIX 'SPICTUR'
C KOL     AN INTEGER, NUMBER OF  COLUMNS  OF THE GIVEN OBJECT
C         MATRIX.
C MAXKOL AN INTEGER .GT. KOL.
C MAXROW AN INTEGER .GE.IROW.
C SPICTUR A REAL 'MAXROW  BY MAXKOL' MATRIX WHOSE FIRST IROW
C         ROWS  AND  FIRST  KOL  COLUMNS  CONTAIN  THE MATRIX
C         'SPICTUR'.
C NSIDE   AN INTEGER, THE SIDE LENGTH  OF THE SQUARE CONVOLU-
C         TION WINDOW. WE TAKE NSIDE=5.
C IMIN, IMAX, JMIN AND JMAX
C         INTEGER  VALUES  WHICH  ARE  THE COORDINATES OF THE
C         RECTANGLE INSIDE WHICH THE OBJECT IS ENCLOSED.
C IMAPL   AN INTEGER INDICATOR SPECIFIED BY THE USER.      IF
C         IMAPL.EQ.1, THIS  SUBROUTINE  MAPS 'SURFACES' : (1)
C         SKEE, (2) SHEE, (3) EDGMAP, (4)  FLATMAP  AND   (5)
```

```
C         ERRSURF  ONTO  'LOGICAL*1' FILES UNDER CERTAIN UNIT
C         NUMBERS (94, 95,... 98 RESPECTIVELY). IF IMAPL.NE.1
C         NO DATA SETS ARE  ASSIGNED TO THESE SURFACES.
C
C      THE OUTPUT OF THE SUBROUTINE.
C SKEE    A REAL MAXROW BY MAXKOL MATRIX. ITS FIRST IROW ROWS
C         AND  FIRST  KOL  COLUMNS  CONTAIN THE VALUES OF THE
C         SMOOTHED GAUSSIAN CURVATURE AT ELEMENS (I,J).
C SHEE    A REAL MAXROW BY MAXKOL MATRIX. ITS FIRST IROW ROWS
C         AND  FIRST  KOL  COLUMNS  CONTAIN THE VALUES OF THE
C         MEAN CURVATURE AT ELEMENTS (I,J).
C EDGMAP A REAL MAXROW BY MAXKOL MATRIX. ITS FIRST IROW ROWS
C         AND  FIRST KOL  COLUMNS  CONTAIN  THE VALUE SQRT(G)
C         =SQRT(1+FU*FU+FV*FV)  AT  ELEMENT  (I,J). THIS IS A
C         MEASURE OF AN EDGE MAGNITUDE AT (I,J).
C FLATMAP A REAL MAXROW BY MAXKOL MATRIX.  ITS  FIRST   IROW
C         ROWS AND FIRST  KOL  COLUMNS  CONTAIN  THE  VALUE :
C         Q  =  (FUU*FUU+2*FUV*FUV+FVV*FVV) AT ELEMENT (I,J).
C         THIS IS A MEASURE OF SURFACE FLATNESS AT (X,Y).
C ERRSURF  A MEASURE OF THE ERROR  IN THE LEAST SQUARES FIT.
C         = A00 - FUU - FVV - SPICTUR(I,J).  IT IS THE RESULT
C         OF THE CONVOLUTION WITH THE 5 BY 5 WINDOW
C
C                      |-65   10    35   10   -65|
C                1     | 10   85   110   85    10|
C               ---    | 35   110 -740  110    35|
C               875    | 10   85   110   85    10|
C                      |-65   10    35   10   -65|
C
C         HOWEVER, INSTEAD, WE  HERE CALCULATE ERRSURF AS THE
C         RESULT OF THE UNSHARP MASK (SQUARES). SEE E.L HALL,
C         COMPUTER IMAGE PROCESSING AND RECOGNITION, ACADEMIC
C         PRESS, 1979, PP. 202-204.
C
C MAPL    AN LOGICAL*1  'MAXROW BY MAXKOL' MATRIX WHOSE FIRST
C         IROW ROWS AND FIRST KOL COLUMNS CONTAIN THE UNFORM-
C         ATED IMAGE STORED UNDER UNIT NUMBER JUNIT.
C ABIGK   MAX |SKEE(I,J)|
C ABIGH   MAX |SHEE(I,J)|
C IERRORC AN INTEGER INDICATOR.  IERRORC=0  INDICATES NORMAL
C         EXIT. IERRORC=-1 INDICATES  PREMATURE EXIT DUE TO A
C         KIND OF ERROR.
C
C      THE MEANING OF OTHER PARAMETERS.
C FU      A REAL MAXROW BY MAXKOL MATRIX. ITS FIRST IROW ROWS
C         AND FIRST KOL COLUMNS CONTAIN  THE  FIRST   PARTIAL
C         DERIVATIVE  DSPICTUR(U,V)/DU.
C FV      A REAL MAXROW BY MAXKOL MATRIX. ITS FIRST IROW ROWS
C         AND FIRST KOL COLUMNS CONTAIN  THE  FIRST   PARTIAL
C         DERIVATIVE  DSPICTUR(U,V)/DV.
C FUV     A REAL MAXROW BY MAXKOL MATRIX. ITS FIRST IROW ROWS
```

```
C         AND FIRST KOL COLUMNS CONTAIN  THE SECOND   PARTIAL
C         DERIVATIVE  D2SPICTUR(U,V)/DUDV.
C FUU     A REAL MAXROW BY MAXKOL MATRIX. ITS FIRST IROW ROWS
C         AND FIRST KOL COLUMNS CONTAIN  THE SECOND   PARTIAL
C         DERIVATIVE  D2SPICTUR(U,V)/DU2.
C FUV     A REAL MAXROW BY MAXKOL MATRIX. ITS FIRST IROW ROWS
C         AND FIRST KOL COLUMNS CONTAIN  THE SECOND   PARTIAL
C         DERIVATIVE  D2SPICTUR(U,V)/DV2.
C EKEE    A REAL MAXROW BY MAXKOL MATRIX. ITS FIRST IROW ROWS
C         AND  FIRST  KOL  COLUMNS  CONTAIN THE VALUES OF THE
C         GAUSSIAN CURVATURE AT ELEMENTS (I,J).
C EHEE    A REAL MAXROW BY MAXKOL MATRIX. ITS FIRST IROW ROWS
C         AND  FIRST  KOL  COLUMNS  CONTAIN THE VALUES OF THE
C         SMOOTHED MEAN CURVATURE AT ELEMENTS (I,J).
C AMAXK,AMAXH,AMAXEDG,AMAXFLAT,AMAXERRS :
C         MAX SKEE(I,J) ,  MAX SHEE(I,J) ,   MAX EDGMAP(I,J),
C         MAX FLATMAP(I,J)  AND MAX ERRSURF(I,J)  RESPECT.
C AMINK,AMINH,AMINEDG,AMINFLAT,AMINERRS :
C         MIN SKEE(I,J) ,  MIN SHEE(I,J) ,   MIN EDGMAP(I,J),
C         MIN FLATMAP(I,J)  AND MIN ERRSURF(I,J)  RESPECT.
C V       A REAL WORKING VECTOR OF DIMENSION MAXROW.
C
C           -------- J (J-2)  (J-1)   J    (J+1)  (J+2)
C          |
C          |(I-2)       V(1)   V(2)   V(3)   V(4)   V(5)
C          |(I-1)       V(6)   V(7)   V(8)   V(9)   V(10)
C          I  I         V(11)  V(12)  V(13)  V(14)  V(15)
C           (I+1)       V(16)  V(17)  V(18)  V(19)  V(20)
C           (I+2)       V(21)  V(22)  V(23)  V(24)  V(25)
C
C
      LOGICAL*1 MAPL(MAXROW,MAXKOL),LOG(4)
      INTEGER*4 ILOG
      EQUIVALENCE (ILOG,LOG(1))
C
      DIMENSION SPICTUR(MAXROW,MAXKOL),FU(MAXROW,MAXKOL),
     1FV(MAXROW,MAXKOL),FUV(MAXROW,MAXKOL),FUU(MAXROW,
     2MAXKOL),FVV(MAXROW,MAXKOL),EKEE(MAXROW,MAXKOL),
     3SKEE(MAXROW,MAXKOL),EHEE(MAXROW,MAXKOL),SHEE(MAXROW,
     4MAXKOL),ERRSURF(MAXROW,MAXKOL),V(MAXROW),EDGMAP
     5(MAXROW,MAXKOL),FLATMAP(MAXROW,MAXKOL)
C
C
      DO 2 J=1,KOL
        DO 1 I=1,IROW
          EHEE(I,J) = 0.0
          SHEE(I,J) = 0.0
          EKEE(I,J) = 0.0
          SKEE(I,J) = 0.0
          EDGMAP(I,J) = 0.0
          FLATMAP(I,J) = 0.0
```

```
            ERRSURF(I,J) = 0.0
    1       CONTINUE
    2     CONTINUE
          AMAX = 1.0E30
          AMAXK = - AMAX
          AMAXH = - AMAX
          AMAXEDG = - AMAX
          AMAXFLAT = - AMAX
          AMAXERRS = - AMAX
C
          AMINK = AMAX
          AMINH = AMAX
          AMINEDG = AMAX
          AMINFLAT = AMAX
          AMINERRS = AMAX
C
          IERRORC = 0
          IF(IROW.GT.MAXROW) GO TO 25
          IF(KOL.GT.MAXKOL) GO TO 26
C  INITIALIZE VARIABLES.
          N2 = NSIDE + NSIDE
          N3 = NSIDE + N2
          N4 = N2 + N2
          N5 = N2 + N3
          NS = NSIDE*NSIDE
          ITRIM = (NSIDE - 1)/2
          ITRIMP1 = ITRIM + 1
          LIMKOL = KOL - ITRIM
          LIMROW = IROW - ITRIM
C CONVOLVING PROCESSES.
          JS = JMIN
          JE = JMAX
          IS = IMIN
          IE = IMAX
          IF(JS.LT.ITRIMP1) JS = ITRIMP1
          IF(IS.LT.ITRIMP1) IS = ITRIMP1
          IF(JE.GT.LIMKOL) JE = LIMKOL
          IF(IE.GT.LIMROW) IE = LIMROW
          DO 5 J=JS,JE
            JM2 = J - 2
            JM1 = J - 1
            JP1 = J + 1
            JP2 = J + 2
            DO 4 I=IS,IE
              IM2 = I - 2
              IM1 = I - 1
              IP1 = I + 1
              IP2 = I + 2
C SEE THE COMMENTS ON V(I).
              V(1) = SPICTUR(IM2,JM2)
              V(2) = SPICTUR(IM2,JM1)
```

```
            V(3) = SPICTUR(IM2,J)
            V(4) = SPICTUR(IM2,JP1)
            V(5) = SPICTUR(IM2,JP2)
            V(6) = SPICTUR(IM1,JM2)
            V(7) = SPICTUR(IM1,JM1)
            V(8) = SPICTUR(IM1,J)
            V(9) = SPICTUR(IM1,JP1)
            V(10) = SPICTUR(IM1,JP2)
            V(11) = SPICTUR(I,JM2)
            V(12) = SPICTUR(I,JM1)
            V(13) = SPICTUR(I,J)
            V(14) = SPICTUR(I,JP1)
            V(15) = SPICTUR(I,JP2)
            V(16) = SPICTUR(IP1,JM2)
            V(17) = SPICTUR(IP1,JM1)
            V(18) = SPICTUR(IP1,J)
            V(19) = SPICTUR(IP1,JP1)
            V(20) = SPICTUR(IP1,JP2)
            V(21) = SPICTUR(IP2,JM2)
            V(22) = SPICTUR(IP2,JM1)
            V(23) = SPICTUR(IP2,J)
            V(24) = SPICTUR(IP2,JP1)
            V(25) = SPICTUR(IP2,JP2)
C
C SUMU, SUMV, SUMUV,SUMUU,SUMVV ARE IX,IY,IXY,IXX,IYY
C   IN THE PAPER OF P.R. BEAUDET, ROTATIONAL INVARIANT
C     IMAGE OPERATORS, ICPR, KYOTO, 1978, PP. 579-583.
C
C AVERAGE IS DENOTED BY SUM.
            SUM = 0.0
            DO 3 K=1,NS
    3         SUM = SUM + V(K)
            SUM = SUM/25.0
C SUMU
            SUMU = 2.0*(V(5) + V(10) + V(15) + V(20) + V(25)
     1             - V(1) - V(6) - V(11) - V(16) - V(21)) +
     2             (V(4) + V(9) + V(14) + V(19) + V(24) -
     3             V(2) - V(7) - V(12) - V(17) - V(22))
            SUMU = XSCALE*SUMU/50.0
C SUMV
            SUMV = 2.0*(V(21) + V(22) + V(23) + V(24) + V(25)
     1             - V(1) - V(2) - V(3) - V(4) - V(5)) +
     2             (V(16) + V(17) + V(18) + V(19) + V(20) -
     3             V(6) - V(7) - V(8) - V(9) - V(10))
            SUMV = YSCALE*SUMV/50.0
C SUMUV
            SUMUV = 4.0*(V(1) + V(25) - V(5) - V(21)) +
     1              2.0*(V(2) + V(6) + V(20) + V(24) - V(4)
     2                   - V(10) - V(16) - V(22)) + (V(7)
     3                     + V(19) - V(9) - V(17))
            SUMUV = XSCALE*YSCALE*SUMUV/100.0
```

```
C SUMUU
            SUMUU = 2.0*(V(1) - V(3) + V(5) + V(6) - V(8) +
     1              V(10) + V(11) - V(13) + V(15) + V(16) -
     2              V(18) + V(20) + V(21) - V(23) + V(25)) -
     3              (V(2) + V(4) + V(7) + V(9) + V(12) +
     4              V(14) + V(17) + V(19) + V(22) + V(24))
            SUMUU = XSCALE*XSCALE*SUMUU/35.0
C SUMVV
            SUMVV = 2.0*(V(1) + V(2) + V(3) + V(4) + V(5) -
     1              V(11) - V(12) - V(13) - V(14) - V(15) +
     2              V(21) + V(22) + V(23) + V(24) + V(25)) -
     3              (V(6) + V(7) + V(8) + V(9) + V(10) +
     4              V(16) + V(17) + V(18) + V(19) + V(20))
            SUMVV = YSCALE*YSCALE*SUMVV/35.0
            SUMU2 = SUMU*SUMU
            SUMV2 = SUMV*SUMV
            SUMUU2 = SUMUU*SUMUU
            SUMVV2 = SUMVV*SUMVV
            SUMUV2 = SUMUV*SUMUV
C
            FU(I,J) = SUMU
            FV(I,J) = SUMV
            FUV(I,J) = SUMUV
            FUU(I,J) = SUMUU
            FVV(I,J) = SUMVV
C UNSHARP MASKING.
            ERRS = SUM - V(13)
            ERRS2 = ERRS*ERRS
            ERRSURF(I,J) = ERRS2
C
C           ERRS = SUM -SUMUU - SUMVV - V(13)
C           ERRS2 = ERRS*ERRS
C           ERRSURF(I,J) = ERRS2
C
            G11 = 1.0 + SUMU2
            G12 = SUMU*SUMV
            G22 = 1.0 + SUMV2
            DETG = 1.0 + SUMU2 + SUMV2
            EDGMAP(I,J) = DETG
            QUE = SUMUU2 + SUMUV2 + SUMUV2 + SUMVV2
            FLATMAP(I,J) = QUE
            SQRTG = SQRT(DETG)
            B11 = SUMUU/SQRTG
            B12 = SUMUV/SQRTG
            B22 = SUMVV/SQRTG
C EKEE IS ITSELF 'K' THE GAUSSIAN CURVATURE AND EHEE
C   IS ITSELF 'H', THE MEAN CURVATURE, SEE  PAGE 58,
C      IN BESL AND JAIN, INVARIANT SURFACE CHARACTERISTICS
C        FOR 3D OBJECT RECOGNITION IN RANGE IMAGES,
C          COMPUTER VISION, GRAPHICS AND IMAGE PROCESSING
C            33(1986), PP. 33-80.
```

```
            EK = (SUMUU*SUMVV - SUMUV*SUMUV)/(DETG*DETG)
            EKEE(I,J) = EK
            SKEE(I,J) = EK
C
            EH = SUMUU + SUMVV + SUMUU*SUMV2 + SUMVV*SUMU2
     *                                  -2.0*SUMU*SUMV*SUMUV
            EH = 0.5*EH/(DETG*SQRTG)
            EHEE(I,J) = EH
            SHEE(I,J) = EH
C
            IF(DETG.LT.AMINEDG) AMINEDG = DETG
            IF(DETG.GT.AMAXEDG) AMAXEDG = DETG
C
            IF(QUE.LT.AMINFLAT) AMINFLAT = QUE
            IF(QUE.GT.AMAXFLAT) AMAXFLAT = QUE
C
            AERRSURF = ERRS2
            IF(ERRS2.LT.AMINERRS) AMINERRS = ERRS2
            IF(ERRS2.GT.AMAXERRS) AMAXERRS = ERRS2
    4       CONTINUE
    5     CONTINUE
C
C SMOOTHING GAUSSIAN AND MEAN CURVATURE
C
      NSMOOTH = 7
      NS2 = NSMOOTH*NSMOOTH
      ITRIM = (NSMOOTH - 1)/2
      ITRIMP1 = ITRIM + 1
      LIMKOL = KOL - ITRIM
      LIMROW = IROW - ITRIM
      JS = JMIN
      JE = JMAX
      IS = IMIN
      IE = IMAX
      IF(JS.LT.ITRIMP1) JS = ITRIMP1
      IF(IS.LT.ITRIMP1) IS = ITRIMP1
      IF(JE.GT.LIMKOL) JE = LIMKOL
      IF(IE.GT.LIMROW) IE = LIMROW
      DO 9 J=JS,JE
        K1 = J - ITRIM
        K2 = J + ITRIM
        DO 8 I=IS,IE
          L1 = I - ITRIM
          L2 = I + ITRIM
          SUMK = 0.0
          SUMH = 0.0
          DO 7 K=K1,K2
            DO 6 L=L1,L2
              SUMK = SUMK + EKEE(L,K)
    6         SUMH = SUMH + EHEE(L,K)
    7         CONTINUE
```

```
            SKEE(I,J) = SUMK/NS2
            SHEE(I,J) = SUMH/NS2
C
            ASKEE = SKEE(I,J)
            IF(ASKEE.LT.AMINK) AMINK = ASKEE
            IF(ASKEE.GT.AMAXK) AMAXK = ASKEE
C
            ASHEE = SHEE(I,J)
            IF(ASHEE.LT.AMINH) AMINH = ASHEE
            IF(ASHEE.GT.AMAXH) AMAXH = ASHEE
    8       CONTINUE
    9     CONTINUE
C
      AA = ABS(AMAXK)
      BB = ABS(AMINK)
      ABIGK = AA
      IF(BB.GT.AA) ABIGK = BB
      EPSK = ABIGK/100.0
C
      AA = ABS(AMAXH)
      BB = ABS(AMINH)
      ABIGH = AA
      IF(BB.GT.AA) ABIGH = BB
      EPSH = ABIGH/100.0
C
      IF(IMAPL.NE.1) GO TO 99
C
C CREATING THE 5 SURFACES SKEE, SHEE, EDGMAP, FLATMAP
C   AND ERRSURF.
C
C FOR DISPLAY PURPOSES WE SCALE THE 5 SURFACES
C   SKEE, SHEE, EDGMAP, FLATMAP AND ERRSURF TO
C     VALUES WHICH RANGE BETWEEN 0.0 AND 255.0.
C
      JUNIT = 94
      REWIND JUNIT
      SCALE = (AMAXK - AMINK)/255.0
      DO 11 J=JS,JE
        DO 10 I=IS,IE
          AK = ABS(SKEE(I,J))
          IF(AK.LT.EPSK) SKEE(I,J) = 0.0
          S = (SKEE(I,J) - AMINK)/SCALE
          ILOG = S
   10     MAPL(I,J) = LOG(1)
   11     CONTINUE
      DO 12 J=1,KOL
   12   WRITE(JUNIT)(MAPL(I,J),I=1,IROW)
      CLOSE(JUNIT)
C
      JUNIT = 95
      REWIND JUNIT
```

```
      SCALE = (AMAXH - AMINH)/255.0
      DO 14 J=1,KOL
        DO 13 I=1,IROW
          AH = ABS(SHEE(I,J))
          IF(AH.LT.EPSH) SHEE(I,J) = 0.0
          S = (SHEE(I,J) - AMINH)/SCALE
          ILOG = S
   13     MAPL(I,J) = LOG(1)
   14     CONTINUE
      DO 15 J=1,KOL
   15   WRITE(JUNIT)(MAPL(I,J),I=1,IROW)
      CLOSE(JUNIT)
C
      JUNIT = 96
      REWIND JUNIT
      SCALE = (AMAXEDG - AMINEDG)/255.0
      DO 17 J=1,KOL
        DO 16 I=1,IROW
          S = (EDGMAP(I,J) - AMINEDG)/SCALE
          ILOG = S
   16     MAPL(I,J) = LOG(1)
   17     CONTINUE
      DO 18 J=1,KOL
   18   WRITE(JUNIT)(MAPL(I,J),I=1,IROW)
      CLOSE(JUNIT)
C
      JUNIT = 97
      REWIND JUNIT
      SCALE = (AMAXFLAT - AMINFLAT)/255.0
      DO 20 J=1,KOL
        DO 19 I=1,IROW
          S = (FLATMAP(I,J) - AMINFLAT)/SCALE
          ILOG = S
   19     MAPL(I,J) = LOG(1)
   20     CONTINUE
      DO 21 J=1,KOL
   21   WRITE(JUNIT)(MAPL(I,J),I=1,IROW)
      CLOSE(JUNIT)
C
      JUNIT = 98
      REWIND JUNIT
      SCALE = (AMAXERRS - AMINERRS)/255.0
      DO 23 J=1,KOL
        DO 22 I=1,IROW
          S = (ERRSURF(I,J) - AMINERRS)/SCALE
          ILOG = S
   22     MAPL(I,J) = LOG(1)
   23     CONTINUE
      DO 24 J=1,KOL
   24   WRITE(JUNIT)(MAPL(I,J),I=1,IROW)
      CLOSE(JUNIT)
```

```
C
      GO TO 99
   25 IERRORC = -1
      WRITE(3,153)IERRORC,IROW
  153 FORMAT(37HO IROW EXCEDES MAXROW:IERRORC,IROW = ,I3,I8)
      GOTO 99
   26 IERRORC = -2
      WRITE(3,154)IERRORC,KOL
  154 FORMAT(37HO  KOL EXCEDES MAXKOL:IERRORC,KOL  = ,I3,I8)
   99 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SGNMAP(IROW,KOL,MAXROW,MAXKOL,MAXRK,SKEE,
     1SHEE,MATSGN,KPTS,XPTS,YPTS,ZED)
C
C     GIVEN ARE THE 'SMOOTHED GAUSSIAN CURVATURE' 'SKEE' AND
C THE  'SMOOTHED MEAN CURVATURE'  'SHEE' AT EACH POINT
(I,J),
C THIS SUBROUTINE CALCULATES  THE  '8'  COMBINATIONS  OF THE
C '+ 0 -' GAUSSIAN CURVATURE AND THE '+ 0 -' MEAN CURVATURES
C AND PRINTS  A  MAP  OF CHARACTERS '12345678' FOR THIS COM-
C BINATION.
C
C     THE INPUT DATA TO THE SUBROUTINE.
C IROW    AN INTEGER: NUMBER OF ROWS OF THE GIVEN IMAGE.
C KOL     AN INTEGER: NUMBER OF COLUMNS OF THE GIVEN IMAGE.
C MAXKOL AN INTEGER .GT. KOL.
C MAXROW AN INTEGER .GE.IROW, MAXROW.GE.MAXKOL.
C MAXRK   AN INTEGER = MAXROW*MAXKOL; 128*128 OR 256*256.
C KPTS    AN INTEGER.   IT IS THE NUMBER OF NONZERO VALUES OF
C         Z-COORDINATED PICTUR(I,J).  THAT IS PICTUR(I,J).GT.
C         EPSZ.
C XPTS    A  REAL  VECTOR  OF LENGTH MAXRK.   ITS  FIRST KPTS
C         ELEMENTS CONTAIN THE X- COORDINATES  OF THE NONZERO
C         Z-COORDINATES OF THE GIVEN OBJECT.
C YPTS    A  REAL  VECTOR  OF LENGTH MAXRK.   ITS  FIRST KPTS
C         ELEMENTS CONTAIN THE Y- COORDINATES  OF THE NONZERO
C         Z-COORDINATES OF THE GIVEN OBJECT.
C SKEE    A REAL MAXROW BY MAXKOL MATRIX. ITS FIRST IROW ROWS
C         AND  FIRST  KOL  COLUMNS  CONTAIN  THE VALUE OF THE
C         SMOOTHED GAUSSIAN CURVATURE AT ELEMENT (X,Y).
C SHEE    A REAL MAXROW BY MAXKOL MATRIX. ITS FIRST IROW ROWS
C         AND  FIRST  KOL  COLUMNS  CONTAIN  THE VALUE OF THE
C         SMOOTHED MEAN CURVATURE AT ELEMENT (X,Y).
C
C     THE OUTPUT OF THE SUBROUTINE.
C MATSGN A CHARACTER*1 MAXROW BY MAXKOL MATRIX.  ITS   FIRST
C         IROW ROWS AND FIRST KOL COLUMNS CONTAIN THE 8-LEVEL
C         SIGN MATRIX VALUES AT (X,Y).
C
      DIMENSION XPTS(MAXRK),YPTS(MAXRK),ZED(MAXRK),
     1SKEE(MAXROW,MAXKOL),SHEE(MAXROW,MAXKOL)
C
      CHARACTER*1 MATSGN(MAXROW,MAXKOL)
C
      CHARACTER STRING*9
      DATA STRING/'12345678 '/
C
  106 FORMAT(30H0  MATRIX MATSGN                )
  128 FORMAT(1X,128A1)
  121 FORMAT(40H0  *****EXIT SUBROUTINE SGNMAP******     )
C
      DO 2 J=1,KOL
```

```
      DO 1 I=1,IROW
    1   MATSGN(I,J) = STRING(9:9)
    2   CONTINUE
      DO 6 K=1,KPTS
        I = XPTS(K)
        J = YPTS(K)
        Z = ZED(K)
        SK = SKEE(I,J)
        SH = SHEE(I,J)
        IF(SH.LT.0.0) GO TO 3
        IF(SH.EQ.0.0) GO TO 4
        IF(SH.GT.0.0) GO TO 5
        GO TO 6
    3   IF(SK.GT.0.0) MATSGN(I,J) = STRING(1:1)
        IF(SK.EQ.0.0) MATSGN(I,J) = STRING(2:2)
        IF(SK.LT.0.0) MATSGN(I,J) = STRING(3:3)
        GO TO 6
    4   IF(SK.EQ.0.0) MATSGN(I,J) = STRING(4:4)
        IF(SK.LT.0.0) MATSGN(I,J) = STRING(5:5)
        GO TO 6
    5   IF(SK.GT.0.0) MATSGN(I,J) = STRING(6:6)
        IF(SK.EQ.0.0) MATSGN(I,J) = STRING(7:7)
        IF(SK.LT.0.0) MATSGN(I,J) = STRING(8:8)
    6   CONTINUE
C
      WRITE(3,106)
      DO 71 I=1,IROW
   71   WRITE(3,128)(MATSGN(I,J),J=1,KOL)
      RETURN
      END
```

```
      SUBROUTINE MOMENTS(NORMSC,MAXRK,MAXX,KPTS,XPTS,
     *YPTS,ZED,VOLUM,XMEAN,YMEAN,AMMT,EPS)
C
C     THIS  SUBROUTINE INPUTS A SET OF  REAL   Z-COORDINATES
C OF THE OBJECT FOR EACH X AND Y COORDINATES. IT OBTAINS THE
C MEAN VALUES OF THE X AND Y OBJECT COORDINATES 'XMEAN'  AND
C 'YMEAN' AND CALCULATES THE 7 NORMALIZED OR   UN-NORMALIZED
C INVARIANT MOMENTS PARAMETERS AMMT(I),I=1:7.
C
C     REFERENCES:
C (1) E. HALL,  COMPUTER  IMAGE  PROCESSING AND RECOGNITION,
C     ACADEMIC PRESS, 1979, PP. 420-421.
C (2) A. GOSHTASBY,  TEMPLATE  MATCHING IN  ROTATED  IMAGES,
C     IEEE TRANS. PATTERN ANALYSIS AND MACHINE INTELLIGENCE,
C     PAMI-7, 338-344, 1985.
C (3) S. MAITRA, MOMENT INVARIANTS, PROCEEDINGS OF THE IEEE,
C     VOL-67, 1979, PP. 697-699.
C
C      INPUT DATA TO TO SUBROUTINE.
C NORMSC AN INTEGER SPECIFIED BY THE USER.  IF NORMSC=0, THE
C        7 CENTRAL MOMENT PARAMETERS ARE NOT INVARIANT UNDER
C        A SCALE CHANGE.  IF NORMSC.NE.0, THEY ARE INVARIANT
C        UNDER A SCALE CHANGE.
C MAXRK  AN INTEGER = MAXROW*MAXKOL; 128*128.
C KPTS   AN INTEGER. IT IS THE NUMBER OF NONZERO ELEMENTS OF
C        MATRIX PICTUR.
C XPTS   A REAL VECTOR  OF  LENGTH  MAXRK.  ITS  FIRST  KPTS
C        ELEMENTS  ARE  THE  X-COORDINATES  OF  THE  NONZERO
C        ELEMENTS OF PICTUR(I,J).
C YPTS   A REAL  VECTOR  OF  LENGTH  MAXRK.  ITS  FIRST  KPTS
C        ELEMENTS  ARE  THE  Y-COORDINATES  OF  THE  NONZERO
C        ELEMENTS OF PICTUR(I,J).
C ZED    AN REAL  VECTOR  OF  LENGTH MAXRK.  ITS FIRST  KPTS
C        ELEMENTS ARE THE NONZERO ELEMENTS OF PICTUR(I,J).
C VOLUM  A REAL NUMBER WHICH IS THE VOLUME OF THE OBJECT.
C XMEAN  A REAL NUMBER WHICH IS THE X COORDINATE OF THE MEAN
C        VALUE OF THE X COORDINATES OF THE SURFACE POINTS.
C YMEAN  A REAL NUMBER WHICH IS THE Y COORDINATE OF THE MEAN
C        VALUE OF THE Y COORDINATES OF THE SURFACE POINTS.
C
C      OUTPUT OF THE SUBROUTINE.
C AMMT   A REAL VECTOR OF LENGTH 7. ITS ELEMENTS ARE THE   7
C        PARAMETERS WHICH ARE  NORMALIZED  INVARIANT CENTRAL
C        MOMENTS FOR THE GIVEN OBJECT.
C
      DIMENSION XPTS(MAXRK),YPTS(MAXRK),ZED(MAXRK),
     *AMMT(MAXX)
C
      PID2 = 2.0*ATAN(1.0)
      EXY = 0.0
      EXX = 0.0
```

```
      EYY = 0.0
      EXXX = 0.0
      EXXY = 0.0
      EXYY = 0.0
      EYYY = 0.0
      DO 1 K=1,KPTS
        Z = ZED(K)
        DX = (XPTS(K) - XMEAN)
        DY = (YPTS(K) - YMEAN)
        EXY = EXY + DX*DY*Z
        EXX = EXX + DX*DX*Z
        EYY = EYY + DY*DY*Z
        EXXX = EXXX + DX*DX*DX*Z
        EXXY = EXXY + DX*DX*DY*Z
        EXYY = EXYY + DX*DY*DY*Z
    1   EYYY = EYYY + DY*DY*DY*Z
C
      C2 = 1.0
      C3 = 1.0
      IF(NORMSC.EQ.0) GO TO 2
      C2 = VOLUM*VOLUM
      C3 = C2*SQRT(VOLUM)
    2 CONTINUE
      EXY = EXY/C2
      EXX = EXX/C2
      EYY = EYY/C2
      EXXX = EXXX/C3
      EXXY = EXXY/C3
      EXYY = EXYY/C3
      EYYY = EYYY/C3
C
      AMMT(1) = EXX + EYY
      AMMT(2) = (EXX - EYY)**2.0 + 4.0*EXY**2.0
      AMMT(3) = (EXXX - 3.0*EXYY)**2 + (3*EXXY - EYYY)**2
      AMMT(4) = (EXXX + EXYY)**2 + (EXXY + EYYY)**2
      AMMT(5) = (EXXX - 3.0*EXYY)*(EXXX + EXYY)*((EXXX +
     1        EXYY)**2 - 3.0*(EXXY + EYYY)**2) + (3*EXXY -
     2        EYYY)*(EXXY + EYYY)* (3*(EXXX + EXYY)**2 -
     3        (EXXY + EYYY)**2)
      AMMT(6) = (EXX - EYY)*((EXXX + EXYY)**2 - (EXXY +
     1        EYYY)**2) + 4*EXY*(EXXX + EXYY)*(EXXY + EYYY)
      AMMT(7) = (3.0*EXXY - EYYY)*(EXXX + EXYY)*((EXXX +
     1        EXYY)**2 - 3.0*(EXXY + EYYY)**2) - (EXXX -
     2        3.0*EXYY)*(EXXY + EYYY)*(3.0*(EXXX +
     3        EXYY)**2 - (EXXY + EYYY)**2)
      DO 3 I=1,7
        AB = ABS(AMMT(I))
    3   IF(AB.LT.EPS) AMMT(I) = 0.0
      RETURN
      END
```